



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**EZ-KET SOFTWARE EDUCATIVO PARA EL APOYO DE
LA ENSEÑANZA DEL IDIOMA INGLÉS A NIVEL KET EN
LA UTM**

**TESIS:
PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

**PRESENTA:
YETNALEZI QUINTAS RUIZ**

**DIRECTOR DE TESIS:
ING. WENDY YANETH GARCÍA MARTÍNEZ**

HUAJUAPAN DE LEÓN, OAX., NOVIEMBRE DE 2005

Índice General

Índice General	i
Índice de Figuras	iii
Índice de Tablas	vi
Introducción	1
1. Software educativo	6
1.1. Bases psicopedagógicas del desarrollo del software educativo.....	8
1.1.1. Dominios del aprendizaje	8
1.1.2. Teorías del aprendizaje.....	9
1.1.3. Modelos de enseñanza para el desarrollo del software educativo.....	11
1.1.4. Clasificación del software educativo.....	15
1.2 Problemas en el desarrollo de un software educativo.....	18
1.2.1. Falta de impacto en los planes de estudio.....	18
1.2.2. Dificultad de estimación de los recursos necesarios para la producción del software educativo.....	19
1.3. Planteamiento de una ingeniería del software educativo.....	19
1.4. Ingeniería del Software: Una base para el desarrollo	20
1.4.1. Proceso de Ingeniería del Software	21
2. La usabilidad y su papel en la calidad del software	24
2.1. Usabilidad.....	25
2.1.1. La importancia de la usabilidad.....	25
2.1.2. Interacción Humano-Computadora (HCI).....	26
2.1.3. Diseño Centrado en el Usuario (UCD).....	28
2.1.4. Métodos utilizados en pruebas de usabilidad	30
3. Análisis y Diseño del Software Educativo	32
3.1. Análisis de requisitos del software educativo EZ-KET	32
3.1.1. Prototipo: Ahorcado	33
3.1.2. Prototipo: Cachar palabras.....	34
3.1.3. Prototipo: Mapa	35
3.1.4. Prototipo: Relacionar imágenes con palabras.....	36
3.1.5. Prototipo: Jugando con los verbos.....	37
3.2. Pruebas de usabilidad	39
3.2.1. Resultado de las pruebas de usabilidad	40
3.3. Análisis de requisitos del sistema de retroalimentación.....	44
3.3.1. Refinamiento de requisitos a través de pruebas de usabilidad en el sistema de retroalimentación	46
3.4. Diseño del Software	46
3.4.1. Diseño del software educativo EZ-KET.....	46

3.4.2. Diseño del sistema de retroalimentación	51
3.4.3. Diseño de la base de datos	55
4. Implementación de EZ-KET	59
4.1 Herramientas de Desarrollo	59
4.2 Software Educativo: EZ-KET	60
4.2.1. Módulo: Registro de Usuario	61
4.2.2. Módulo: Inicio de Sesión.....	62
4.2.3. Módulo: Recordar Contraseña.....	65
4.2.4. Módulo: Cambiar Información del Estudiante	66
4.2.5. Módulo: Ayuda.....	66
4.2.6. Juego: Ahorcado.....	67
4.2.7. Juego: Relacionar Imágenes con Palabras.....	71
4.2.8. Juego: Cachar Palabras.....	72
4.2.9. Juego: Mapa.....	73
4.2.10. Juego: Jugando con los Verbos	74
4.2.11. Juego: Describe, localiza partes y objetos de una casa.....	76
4.2.12. Ejercitadores tipo A y tipo B	78
4.3 Sistema de Retroalimentación	80
4.3.1. Módulo: <i>New Topic</i>	81
4.3.2. Módulo: <i>Modify a Topic</i>	82
4.3.3. Módulo: <i>New Item</i>	83
4.3.4. Módulo: <i>Modify an Item</i>	84
4.3.5. Módulo: <i>Create a New Question for the Verbs Game</i>	85
4.3.6. Módulo: <i>Modify a Question for the Verbs Game</i>	86
4.3.7. Módulos: <i>Create a New Exercise type A and type B</i>	86
4.3.8. Módulos: <i>Modify an exercise type A and modify an exercise type B</i>	88
4.3.9. Módulo: <i>Report</i>	89
4.3.10. Módulo: <i>Help</i>	91
5. Resultados.....	92
5.1. Comparación del desempeño.....	95
Conclusiones y Trabajos Futuros.....	99
Bibliografía.....	103
Anexo A. Temario.....	109
Anexo B. Reportes	111
Anexo C. Manual de usuario	117
Anexo D. Manual de instalación.....	133

Índice de Figuras

Figura 1.1 Etapas del proceso de enseñanza, según R. Gagné	13
Figura 1.2 Capas de la Ingeniería de Software	20
Figura 1.3 Modelo de cascada con prototipado	22
Figura 2.1. Esquema de HCI	27
Figura 2.2. Proceso iterativo de la Metodología Centrada en el Usuario	30
Figura 3.1. Primer prototipo del juego del ahorcado	33
Figura 3.2. Primer prototipo del juego de cachar palabras	34
Figura 3.3. Primer prototipo del juego del mapa	35
Figura 3.4. Primer prototipo del juego de relacionar palabras con imágenes	36
Figura 3.5. Primer prototipo del juego de verbos	37
Figura 3.6. Prueba de usabilidad en curso.	40
Figura 3.7. Gráfica comparativa de la tarea registrar usuario.	41
Figura 3.8. Gráfica correspondiente a la tarea registrar usuario.	41
Figura 3.9. Gráfica comparativa de la tarea recordar contraseña.	42
Figura 3.10. Gráfica correspondiente a la tarea recordar contraseña.	42
Figura 3.11. Gráfica comparativa de la tarea entrar al juego y utilizarlo.	43
Figura 3.12. Gráfica correspondiente a la tarea entrar al juego y utilizarlo.	44
Figura 3.13. Primer prototipo para agregar un tema en el sistema de retroalimentación.	45
Figura 3.14. Primer prototipo del sistema de retroalimentación.	45
Figura 3.15. Diagrama de casos de uso del software educativo de alumnos.	47
Figura 3.16. Diagrama de casos de uso del sistema de retroalimentación.	52
Figura 3.17. Diagrama lógico de la base de datos	56
Figura 4.1. Estructura del software educativo EZ-KET	61
Figura 4.2. Registro del estudiante.	62
Figura 4.3. Inicio de sesión	62
Figura 4.4. Juego Serpientes y Escaleras	63
Figura 4.5. Juego Serpientes y Escaleras para KET 1 versión final	64
Figura 4.6. Juego Serpientes y Escaleras para KET 2 versión final	64

Figura 4.7. Juego Serpientes y Escaleras para KET 3 versión final.....	65
Figura 4.8. Recordar contraseña.....	65
Figura 4.9. Interfaz para cambia el nivel KET y/o grupo.....	66
Figura 4.10. Ayuda implementada en el sistema EZ-KET.....	67
Figura 4.11. Juego del ahorcado para KET 1.....	68
Figura 4.12. Juego del ahorcado para KET 2.....	69
Figura 4.13. Juego del ahorcado para KET 3.....	70
Figura 4.14. Juego de relacionar imágenes con palabras.....	71
Figura 4.15. Juego de cachar palabras.....	72
Figura 4.16. Juego del mapa para KET 1 y KET 2.....	73
Figura 4.17. Juego del mapa para KET 3.....	74
Figura 4.18. Jugando con verbos para KET 2 y KET 3.....	75
Figura 4.19. Jugando con verbos para KET 2 y KET 3.....	75
Figura 4.20. Juego de la casa para KET 2.....	77
Figura 4.21. Juego de la casa para KET 2 versión final.....	77
Figura 4.22. Ejercitador tipo A.....	78
Figura 4.23. Ejercitador tipo B.....	78
Figura 4.24. <i>Splash screen</i>	79
Figura 4.25. Ventana previa a la versión final.....	80
Figura 4.26. Ventana de aviso final.....	80
Figura 4.27. Interfaz principal del sistema de retroalimentación.....	81
Figura 4.28. Pantalla para agregar un tema.....	81
Figura 4.29. Pantalla para modificar un tema.....	82
Figura 4.30. Pantalla para agregar un elemento asociado a un tema.....	83
Figura 4.31. Pantalla para modificar un elemento asociado a un tema.....	84
Figura 4.32. Pantalla para agregar una pregunta u oración de verbos.....	85
Figura 4.33. Pantalla para modificar una pregunta u oración de verbos.....	86
Figura 4.34. Pantalla para agregar un ejercitador tipo A.....	87
Figura 4.35. Pantalla para agregar un ejercitador tipo B.....	87
Figura 4.36. Pantalla para modificar un ejercitador tipo A.....	88
Figura 4.37. Pantalla para modificar un ejercitador tipo B.....	89

Figura 4.38. Pantalla para seleccionar un reporte.....	90
Figura 4.39. Reporte individual.....	90
Figura 4.40. Ejemplo de pantalla de ayuda para el sistema de retroalimentación.....	91
Figura 5.1. Gráfica comparativa de porcentajes de aciertos para KET 1.	95
Figura 5.2. Gráfica comparativa de porcentajes de aciertos para KET 2.	96
Figura 5.3. Gráfica comparativa de porcentajes de aciertos para KET 3.	97

Índice de Tablas

Tabla 1.1 Dominios del aprendizaje	9
Tabla 3.1. Requisitos identificados en cada juego.....	38
Tabla 5.1. Resultados de la evaluación tradicional	93
Tabla 5.2. Resultados obtenidos con el sistema	94
Tabla 5.3. Comparación de porcentajes de aciertos KET 1.....	95
Tabla 5.4. Comparación de porcentajes de aciertos KET 2.....	96
Tabla 5.5. Comparación de porcentajes de aciertos KET 3.....	97

Introducción

Aprender es una característica propia de cada individuo que contribuye a fundamentar las bases del desarrollo humano. Durante el aprendizaje, el individuo adquiere el conocimiento por medio del estudio, la ejercitación o la experiencia. A medida que evolucionamos como civilización, surgen nuevas áreas que se dedican al estudio del comportamiento y aprendizaje humano, tratando de mejorar el proceso de enseñanza/aprendizaje en todos los aspectos de la actividad humana.

El mejoramiento de este proceso a través del refinamiento de las técnicas de aprendizaje, permite que las nuevas generaciones incorporen herramientas que no existían hace algunos años, como es el uso de la computadora en el modelo enseñanza/aprendizaje. El alto grado de tecnología involucrado en el proceso de enseñanza es una consecuencia de la evolución hacia una sociedad que requiere la integración de la tecnología en los diferentes procesos que realiza.

Los sistemas de enseñanza parten de métodos originados en diversas corrientes ideológicas que pretenden formalizar una metodología adecuada a las necesidades. El objetivo de un modelo educativo está relacionado con la asimilación adecuada y precisa de los conocimientos (a partir del razonamiento), que ayude al educando a comprender un fenómeno.

Un software educativo que hace uso de la computadora mejora el proceso de enseñanza/aprendizaje, esto lo hace una herramienta competitiva dado que el uso de nuevas tecnologías y el diseño interactivo de materiales didácticos permiten una relación básica entre los diferentes elementos que integran cualquier proceso formativo. Estos elementos son: el estudiante, el profesor y los materiales complementarios [URL11].

Hoy en día, el uso del idioma Inglés se ha vuelto necesario para la comunicación personal, ya sea en el campo educacional, profesional, laboral o comercial, siendo la escuela la forma básica y original en métodos de enseñanza, utiliza la postura tradicional profesor-alumno.

La Universidad Tecnológica de la Mixteca (UTM) cuenta con un Centro para la Enseñanza del Idioma Inglés y entre su material didáctico dispone de software especializado, éste se encuentra en una Sala de Autoacceso para apoyar al alumno autodidacta. Sin embargo, dicho software no cumple los requerimientos curriculares de los niveles KET¹, PET² y POST PET³. En dicha sala se encuentra software como: *Rosetta Stone*, *New Interchange*, *Focus on Grammar Basic* y *Focus on Grammar Intermediate*, los cuales hacen uso de una gran variedad de imágenes y sonido para la enseñanza del idioma; *Hablemos Inglés*, *Making Conections* y *ES-LAB* son programas que proporcionan al alumno ejercicios de repaso; *Passport* es un diccionario interactivo en el que se puede escuchar la pronunciación y visualizar la forma correcta de escribir una palabra; *Learn to Speak English* se enfoca principalmente al área de pronunciación, aunque incluye algunas lecciones de gramática. De estos sistemas, los que realizan una evaluación al terminar un

¹ Key English Test

² Preliminary English Test

³ Post Preliminary English Test

tema son: *Focus on Grammar Basic*, *Focus on Grammar Intermediate*, *New Interchange*, *ES-LAB* y *Learn to Speak English*.

Ninguno de estos programas permite agregar nueva información o mantener un registro del avance del usuario, éstos fueron diseñados para interactuar con un solo usuario. Por lo que al querer utilizarlos como herramientas educativas, su funcionalidad disminuye. Por ejemplo, para usar el software *Rosetta Stone* en la Sala de Autoacceso de la UTM, en cada una de las máquinas se crean cuentas diferentes. En caso de que el alumno utilice una computadora que no le fue asignada, por lo tanto es una aplicación local.

Dado lo anterior, este trabajo de tesis propone:

Hipótesis

A través de un software educativo se pretende apoyar a estudiantes en nivel KET específicamente a que mejoren y refuercen su comprensión del idioma Inglés en las áreas de: vocabulario, comprensión gramatical y auditiva.

Objetivo General

Implementar y desarrollar un software educativo basado en un modelo formal de enseñanza bajo un diseño centrado en el usuario, que ayude al alumno a reforzar y mejorar (de una manera amena) su conocimiento del idioma inglés; diseñado específicamente para las necesidades de los alumnos de la UTM y apegado a los temarios establecidos por el centro de idiomas de la universidad.

Objetivos Específicos

Generar un software educativo que apoye a los estudiantes a nivel KET a mejorar el idioma Inglés

Proporcionar una interfaz amigable que cumpla con el proceso de diseño centrado en el usuario

Aplicar un modelo formal de enseñanza

Seguir un modelo formal de desarrollo de software

Desarrollar un sistema de retroalimentación para asegurar la vigencia de los datos

Alcances

El software proporcionará un avance progresivo del usuario controlado por resultados

El software se apegará al temario de los tres niveles de KET existentes en la UTM

La información de los usuarios será almacenada en una BD residente en un servidor

El acceso de cada usuario al software estará controlado por contraseña para garantizar la confidencialidad de la información

La modalidad principal del software educativo será el juego

Los profesores podrán modificar el contenido del software de acuerdo a sus necesidades

Limitaciones

Dos usuarios o más no podrán competir uno contra el otro, pero podrán usar el sistema al mismo tiempo.

El sistema abarcará únicamente los temas correspondientes a los tres niveles KET existentes en la UTM.

El sistema soportará modificaciones por parte de los profesores siempre y cuando éstas no atenten contra la estructura fundamental del software, es decir, será posible modificar el contenido respetando los tres niveles de KET, la forma de presentación de datos, así como los juegos incorporados en el mismo.

Esta tesis consta de los siguientes capítulos:

Capítulo I. Software Educativo: Marco teórico donde se destacan las bases psicopedagógicas del desarrollo del software educativo, los modelos de enseñanza, su

clasificación; los modelos del proceso de ingeniería de software, así como los modelos de proceso de desarrollo.

Capítulo II. La Usabilidad y su papel en la calidad del software. En este capítulo se consideran los conceptos básicos de la usabilidad, la importancia de la misma en el desarrollo del software, la interacción humano-computadora y la metodología de diseño centrado en el usuario.

Capítulo III. Análisis y Diseño del Software Educativo. Se presenta el análisis de requisitos, el diseño del sistema mediante diagramas de casos de uso, el modelo de prototipos, el diseño de la base de datos y las pruebas de usabilidad.

Capítulo IV. Desarrollo e Implementación de EZ-KET. En este capítulo se consideran los aspectos relacionados a la implementación del sistema, como es la descripción de cada uno de los módulos que lo integran, las pruebas de usabilidad utilizadas, así como la descripción del software educativo y la descripción del software de retroalimentación.

Capítulo V. Resultados. Se muestran los resultados obtenidos de la interacción con el sistema, los cuales son comparados con los obtenidos de forma tradicional mediante la aplicación de un pequeño examen escrito.

Finalmente se presentan las *Conclusiones y Trabajos Futuros*. Se incluyen opciones de crecimiento y mejora del sistema que pueden realizarse en versiones siguientes.

Capítulo 1

1. Software educativo

La importancia del software educativo en la actualidad, sus características, la problemática vinculada a su desarrollo y los modelos psicopedagógicos que los sustentan, son algunos de los puntos tratados en este capítulo.

Se denomina *software educativo* a los “programas creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y aprendizaje” [URL07]. Todo esto implica que de alguna manera, parte del contenido de aprendizaje que se pretende lograr estará sustentado, implícita o explícitamente en el software.

Un software educativo debe cumplir con una serie de características funcionales, técnicas y pedagógicas, que son [URL04]:

Facilidad de uso e instalación.

Versatilidad: adaptación a entornos, estrategias didácticas y usuarios.

Calidad del entorno visual: diseño general claro, atractivo de las pantallas, calidad técnica.

Calidad en los contenidos: la información que se presenta es correcta y actual.

Navegación e interacción: buena estructuración del programa, un entorno transparente que permita al usuario tener el control, gestión de preguntas, respuestas y acciones.

Originalidad y uso de tecnología avanzada.

Capacidad de motivación.

Adecuación a los usuarios y a su ritmo de trabajo.

Potencialidad de los recursos didácticos.

Fomento a la iniciativa y al autoaprendizaje.

No todas las características anteriores deben cumplirse a la perfección, cada una se desarrolla con propósitos específicos. Su base tecnológica corresponde a lo que en ese momento existe en el mercado. Así, los programas computacionales son elaborados y diseñados con lógicas y objetivos propios [URL09].

En algunas ocasiones, el software educativo está relacionado con el desarrollo del *courseware*, por lo tanto, es importante conocer las diferencias entre ambos. Se concibe el *courseware* [GJ02] como un sistema que promueve una acción formativa, con el apoyo de la tecnología computacional. Su objetivo es, por tanto, la realización de una serie de actividades y la integración de un conjunto de medios para alcanzar un determinado nivel de aprendizaje por parte de los estudiantes (un subsistema software). El resultado del *courseware* puede incluir muchos otros elementos, además del subsistema software; por ejemplo, documentos en diversos formatos y soportes (audio, vídeo, papel), así como

métodos, procedimientos o técnicas orientadas a promover y/o controlar determinadas variables del proceso de enseñanza.

Desarrollar un sistema tipo *courseware* implica cambios en los métodos de enseñanza, además de desarrollar un software educativo.

1.1. Bases psicopedagógicas del desarrollo del software educativo

El software educativo se destina a apoyar el proceso de enseñanza-aprendizaje de un determinado contenido. Por ello, es importante tener en cuenta los aspectos pedagógicos involucrados en su desarrollo.

La pedagogía se contempla como la ciencia que explica el hecho educativo y como la tecnología que apoyándose en principios, leyes, experiencias e investigaciones, se convierte en un saber fundamentado y generalizable que evita y se aleja del azar [GL89].

1.1.1. Dominios del aprendizaje

De acuerdo a Shaffer [SD00] el *aprendizaje* es un cambio relativamente permanente del comportamiento producido por la experiencia (repetición, práctica, estudio u observación) más que por herencia. Otra definición dada por Alonso [AC95] dice que “el aprendizaje es el proceso de adquisición de una disposición, relativamente duradera, para cambiar la percepción o la conducta como resultado de una experiencia”.

El uso de la computadora para intensificar el aprendizaje requiere comprender en qué consiste, cómo se explica, los factores que lo afectan y qué se puede obtener de él, todo ello orientado a instrumentar entornos de aprendizaje soportados en tecnologías que respondan a las características del que aprende y de lo que intenta aprender [GJ02].

El aprendizaje implica cambios en habilidades, actitudes, conductas, marcos de referencia conceptuales, conocimientos, modelos mentales y/o destrezas. Estos cambios

tienden a persistir a lo largo del tiempo, aunque algunas veces lo aprendido se desvanece o se sustituye por conocimientos nuevos.

Las investigaciones de Bloom y de Gagné señalan cinco dominios en los que se manifiesta el aprendizaje, éstos son: cognitivo, afectivo, motor, psicomotor y metacognitivo [BB56] [GR92]. La Tabla 1.1 muestra la descripción de los aspectos vinculados a cada uno de ellos.

Tabla 1.1 Dominios del aprendizaje [GR92]

Dominios del aprendizaje	Aspectos con los que se relaciona
Cognitivo	Procesos de pensamiento, de apropiación del conocimiento
Afectivo (Actitudinal)	Desarrollo de sentimientos, actitudes y valores
Motor	Aprendizaje de movimientos físicos
Psicomotor	Procesos mentales implicados en movimientos físicos que pueden llegar a ser realizados de manera automática.
Metacognitivo	Estrategias para realizar tareas de aprendizaje que se usan sin necesidad de pensar acerca de ellas

El diseño de cualquier acción orientada a promover el aprendizaje se debe centrar en la producción de uno o más de estos dominios. En el caso de uso de tecnologías basadas en computadoras, se debe procurar que el aprendiz tenga una apreciación positiva a éstas tecnologías, lo cual se logra mediante una interfaz de usuario agradable e intuitiva.

1.1.2. Teorías del aprendizaje

A lo largo del tiempo se han desarrollado diferentes teorías del aprendizaje. Estas se orientan a describir cómo se aprenden nuevas ideas y conceptos. Las teorías del aprendizaje que predominan hoy en día son: conductismo, cognitivismo y constructivismo, cuyas características generales se describen a continuación:

Conductismo: Considera que el aprendizaje da por resultado cambios observables en la conducta del sujeto. Se enfoca hacia la repetición de patrones de conducta, hasta que éstos se realicen de manera automática. Sostiene que la conducta humana consiste en un gran conjunto de reflejos innatos y adquiridos. Algunos autores relevantes en el desarrollo de esta teoría han sido: Watson, Skinner, Pavlov, Thorndike y Gagné [SP90].

Cognitivism: Considera que el aprendizaje ocurre cuando los aprendices son capaces de incorporar nuevos conceptos e ideas a su estructura cognitiva, al reconocer una relación entre algo que ya conocen y aquello que están aprendiendo. Los cambios en la conducta les sirven como indicadores para entender lo que está pasando en la mente del aprendiz. [AD68] [GT98].

Constructivismo: Se sustenta en la premisa de que cada persona construye su propia perspectiva del mundo que lo rodea a través de sus propias experiencias y esquemas mentales desarrollados. Se enfoca hacia la preparación del aprendiz para resolver problemas en condiciones ambiguas. Los pioneros de esta teoría fueron Barlett y Piaget [URL06] [MM91].

Tanto el conductismo como el cognitivism, se basan en datos objetivos acerca de los cambios en la conducta del aprendiz. Una diferencia entre ambas es que para los conductistas la mente humana es una “caja negra”, mientras que los cognitivistas intentan verla más como una “caja transparente”.

Ninguna de estas teorías se debe considerar universalmente válida frente a las demás, ya que, existen aspectos ambientales, socioculturales y técnicos que pueden influir a la hora de tomar como base alguna de ellas en el diseño de algún sistema educativo.

El software educativo que se presenta en esta tesis se basa en el constructivismo y en el conductismo, ya el usuario tiene que contestar y el sistema asigna una puntuación. Además, el usuario decide qué juegos realizar y cuáles no.

Richards y Rodgers [RJ98] mencionan tres teorías distintas sobre la lengua y la naturaleza del conocimiento lingüístico que inspiran de manera implícita o explícita los enfoques, los métodos actuales de enseñanza de idiomas. Estas teorías son:

Estructural: Considera la lengua como un sistema de elementos relacionados estructuralmente para codificar el significado. Se piensa que el objetivo del aprendizaje de una lengua es el conocimiento de los elementos del sistema, generalmente se definen como unidades fonológicas, unidades gramaticales, operaciones gramaticales y elementos léxicos.

Funcional: Considera a la lengua como un vehículo para la expresión de un significado funcional. Esta teoría hace hincapié en la dimensión semántica y comunicativa.

Interactiva: Considera a la lengua como un vehículo para el desarrollo de relaciones personales y la realización de transacciones de tipo social entre individuos. Se concibe como un instrumento para crear y mantener relaciones sociales. Las áreas de investigación en las que se centra el enfoque interactivo para la enseñanza de la lengua incluyen el análisis de la interacción y el análisis de la conversación. La teoría interactiva se centra en el análisis de la estructura de modelos de movimientos, actos, negociación e interacción que se encuentra en los intercambios conversacionales.

Estas teorías ayudan a seleccionar los contenidos, que es la forma en que se presenta la información a los alumnos, además proporcionan una idea general de cómo imparten clases los profesores en la enseñanza de un idioma.

1.1.3. Modelos de enseñanza para el desarrollo del software educativo

Es necesario partir de un concepto claro sobre qué es un modelo para poder de definir los modelos de enseñanza que existen para el desarrollo del software educativo.

Modelo se puede definir como la “representación de un sistema o realidad compleja que se elabora para facilitar su comprensión y estudio” [URL10]. Para diseñar un software educativo, es importante basarse en un modelo que describa el proceso de enseñanza. A continuación se describen los dos modelos más reconocidos y aceptados en el diseño del software educativo.

a) **Modelo basado en nueve etapas de instrucción.** Este modelo fue propuesto por Gangé [GR92]. Constituye un modelo general de enseñanza, el cual es aplicable en el diseño instruccional de cualquier actividad formativa, cuyas etapas son:

0. ***Atraer la atención:*** La atención del estudiante puede ser captada a través de estímulos logrados con la ayuda de diversos medios; por ejemplo: videos, audio, animación, gráficos, el uso de diferentes combinaciones de colores, al variar el tamaño y tipo de la fuente del texto; se deben destacar los aspectos relevantes.
0. ***Informar al aprendiz sobre el objetivo:*** Ayuda al estudiante a esclarecer el resultado final de su interacción con la computadora, a partir del punto específico donde se encuentra.
0. ***Recordar los aprendizajes previos:*** El aprendizaje es más eficiente cuando el programa estimula a recordar información y destrezas relevantes antes de presentar nuevos contenidos.
0. ***Presentar información estimulante:*** Implica la utilización de estrategias, técnicas y materiales que faciliten el dominio de los objetivos. Comenzar presentando algunos conceptos simples y proceder luego hacia otros más elaborados.
0. ***Guiar hacia el objetivo:*** Contribuye a la eficacia del aprendizaje. La magnitud de la orientación o guía debe variar en dependencia del contenido que se aborde.
0. ***Provocar la actuación práctica:*** Logra la retención y el dominio de la información. Se encamina a conseguir que el estudiante recuerde lo aprendido y lo practique.

- 0. ***Evaluar el desempeño***: Permite activar la recuperación de información y refuerza la memoria. Sirve para medir el desempeño en términos de un conjunto de criterios asociados a los objetivos.
- 0. ***Informar acerca del desempeño***: La retroalimentación es proveer información a los estudiantes acerca de la corrección de su desempeño. Es importante aclarar que proveer retroalimentación no es lo mismo que calificar. El hecho de que el estudiante pueda conocer de inmediato acerca de la corrección o error en cualquiera de sus respuestas, contribuye a mantener su motivación hacia el aprendizaje.
- 0. ***Consolidar la retención y transparencia***: En lo que respecta a la retención adecuada de los conceptos y reglas, la planificación del proceso de enseñanza debe incluir revisiones sistemáticas, espaciadas a lo largo de semanas o meses.

La Figura 1.1 muestra en forma gráfica las etapas del proceso de enseñanza.

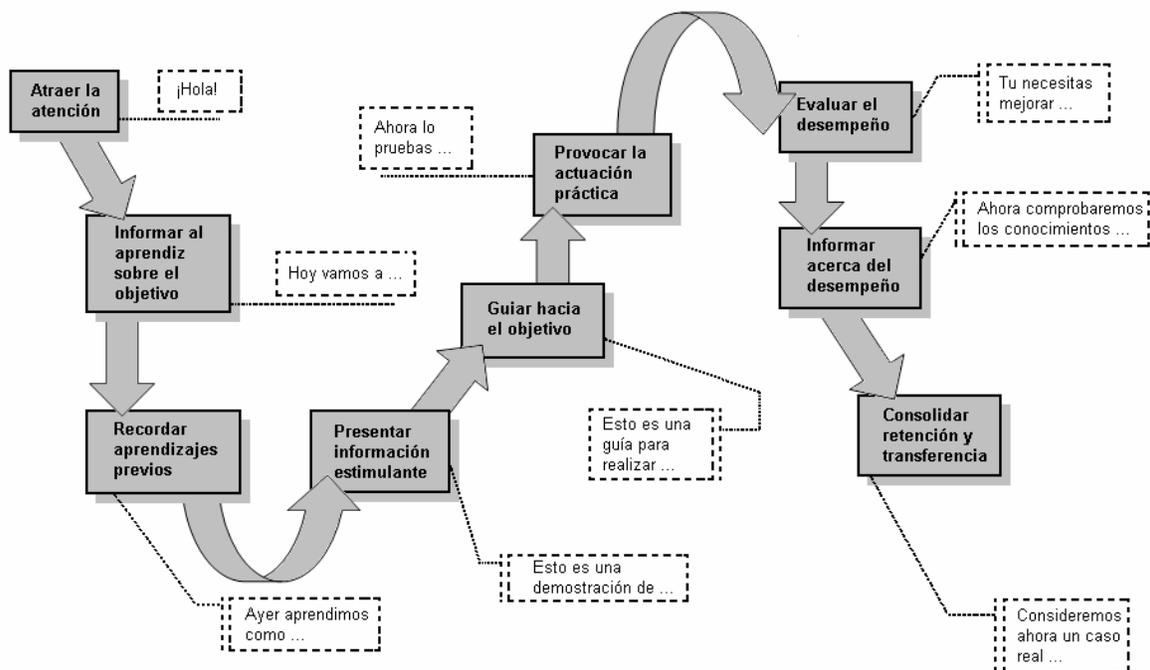


Figura 1.1 Etapas del proceso de enseñanza, según R. Gagné [GR92].

) **Modelo de cuatro etapas del proceso de enseñanza.** Un modelo para lograr una instrucción exitosa, que toma en cuenta los resultados de numerosos estudios y experiencias psicopedagógicas, es el propuesto por Alessi y Tollip [AS00]. Según este modelo, en todo proceso de enseñanza es necesario tener en cuenta cuatro etapas:

0. ***Presentación de la información o contenidos:*** Para enseñar algo nuevo, se debe mostrar primero la información. Para la información verbal o visual, un profesor puede valerse de normas y ejemplos, mostrar dibujos o proporcionar información no hablada. La presentación de la información se puede hacer también con la ayuda de algún medio didáctico y no sólo a través de personas, como es el caso de muchos entornos computacionales destinados a enseñar sin la participación presencial del profesor.
0. ***Guía al estudiante a través de la información:*** Mientras que la primera fase, la presentación de la información se centra en el profesor o en el medio didáctico, esta segunda fase es más interactiva e incluye al estudiante y profesor. En esta fase, el profesor observa al estudiante, corrige sus errores y le proporciona sugerencias o consejos.
0. ***Realización de prácticas en base a los conocimientos adquiridos:*** Esta fase se centra en el estudiante. Aunque el profesor o algún medio interactivo pueden observar, guiar y corregir su actuación, el énfasis en la realización de actividades prácticas recae sobre el estudiante.
0. ***Evaluación del proceso de enseñanza aprendizaje:*** No se debe suponer que la instrucción resultará exitosa para todos los estudiantes, motivo por el cual, es necesario evaluar el aprendizaje logrado. Esta evaluación proporcionará información acerca del nivel de aprendizaje, así como la calidad y las necesidades futuras de enseñanza.

Este modelo de enseñanza es muy general y de acuerdo a las diferentes teorías del aprendizaje, (descritas en el Apartado 1.1.2), se puede adaptar a algún concepto en

particular de cómo llevar a cabo la enseñanza. Se hace necesario recordar que el software educativo no necesariamente tiene que soportar o apoyar a las cuatro etapas mencionadas.

Esta tesis adopta el modelo de cuatro etapas del proceso de enseñanza como marco referencial del proceso de enseñanza, debido a que éste modelo se sustenta en resultados de investigaciones acerca de la enseñanza impartida por profesores. Además, que las etapas incluyen de forma resumida al modelo de nueve etapas.

1.1.4. Clasificación del software educativo

Los programas computacionales se han diseñado para ciertas tareas en particular, lo que ha permitido clasificarlos en diversos grupos. Considerando las etapas del proceso de enseñanza y las teorías del aprendizaje que resultan afines, se distinguen diversas modalidades del software educativo, las cuales son [GJ02]:

- Tutoriales
- Hipermedia
- Ejercitadores
- Simulaciones
- Juegos
- Herramientas
- Evaluaciones
- Aprendizaje soportado en la web

A continuación se describen de forma general cada una de ellas.

Tutoriales

Gran parte del software educativo existente corresponde a esta categoría. Se trata de programas que intentan reproducir la forma de enseñanza conocida como “diálogo socrático”, le presentan información al estudiante y lo guían en las etapas iniciales de adquisición del conocimiento, a través de preguntas y analizando sus respuestas, con el fin

de provocar la reflexión del estudiante y promover el aprendizaje de los conceptos de estudio.

La actividad del estudiante está esencialmente dirigida por el programa. La base psicopedagógica fundamental de este tipo de software educativo es la teoría del conductismo, aunque también se aprecia una tendencia a la incorporación de ideas provenientes de la teoría del cognitivismo [GJ02].

Hipermedia

Un hipertexto se puede definir como un grafo cuyos nodos se relacionan mediante vínculos, que permiten la organización no lineal de la información. En una estructura hipermedia los nodos pueden contener información expresada en diferentes formatos (texto, gráfico, sonido, video, animación) [NJ95].

Los fundamentos psicopedagógicos de la Hipermedia educativa pueden relacionarse tanto con la teoría del conductismo como con la del constructivismo, en dependencia del grado de iniciativa que esté a disposición del estudiante.

Ejercitadores

El propósito fundamental de los ejercitadores no es enseñar nuevos contenidos, sino repetir el material a ser aprendido hasta que el estudiante demuestre que lo domina. Los ejercitadores, al igual que los tutoriales convencionales, incorporan como base teórica los principios del conductismo.

Simulaciones

Instrumenta el modelo de algún fenómeno o actividad, acerca del cual se pretende que el estudiante aprenda, a través de la interacción con el programa. Permite al estudiante operar libremente, dentro de los límites de un determinado entorno o micromundo. Las simulaciones se pueden aplicar bajo un enfoque objetivista o un enfoque constructivista.

Juegos

Los juegos didácticos, se pueden combinar con ejercitadores o simulaciones. Pueden servir como un entorno de aprendizaje por descubrimiento, o se pueden utilizar para integrar aprendizaje sobre un conjunto de materiales, como se hace a menudo con los juegos del género de aventura.

Herramientas

Consisten en programas de computadora que los estudiantes pueden utilizar, conjuntamente con otros medios o actividades para alcanzar alguna meta educacional.

Evaluaciones

El software educativo puede ser empleado para facilitar: la elaboración de exámenes, la administración de pruebas y exámenes.

Aprendizaje soportado en la web

Se utiliza principalmente en combinación con la modalidad de Hipermedia educativa. La web sirve para facilitar el acceso a la información y como medio de distribución.

Es importante resaltar que la clasificación presentada anteriormente se basa sólo en criterios pedagógicos; la separación entre dichas categorías no siempre es exclusiva.

El software educativo que se desarrolla en esta tesis se basa en la combinación de diferentes modalidades que son: juegos, herramientas y ejercitadores, enfocado a estudiantes de formación profesional. En la modalidad de juego, el estudiante practica los conocimientos de manera no repetitiva, permitiendo el aprendizaje por descubrimiento. La modalidad de herramienta se logra a través de la incorporación de mecanismos que permiten a los profesores modificar de forma abierta y flexible los contenidos temáticos. Finalmente, en la modalidad de ejercitador, se refuerzan los conocimientos mediante la aplicación práctica de los mismos.

1.2 Problemas en el desarrollo de un software educativo

El software educativo abarca una gran gama de subdominios, tipos de sistemas de software, requisitos e idiosincrasias, lo suficientemente diversa como para justificar el estudio de los principios, métodos y herramientas de la ingeniería del software que mejor se vinculen a su desarrollo, así mismo, cómo particularizarlos y adaptarlos a este ámbito. De acuerdo a J. González [GJ02], se pueden identificar dos problemas fundamentales que afectan al software educativo actualmente:

Falta de impacto en los planes de estudio.

Dificultad de estimación de los recursos necesarios para su producción.

En los siguientes apartados se describe con detalle cada uno de ellos.

1.2.1. Falta de impacto en los planes de estudio

Los inicios de la computación educativa estaban llenos de profecías sobre el gran potencial de la instrucción basada en la computadora. En los últimos se han denunciado la falta de impacto que tiene el software educativo en la enseñanza.

Por ejemplo, Roschelle y Kaput [RJ96] reclaman la integración de innovaciones en la investigación educativa como un conjunto de herramientas de estudio. Consideran que, frente a la existencia de lo que ellos denominan una arquitectura basada en “aplicaciones isla”, se requiere introducir la integración como un elemento crítico de la tecnología educativa. Otros autores argumentan que la integración de los módulos software y los planes de estudios es crucial para la adopción a gran escala de la tecnología en las escuelas y que es de hecho, esta falta de integración la culpable en gran medida del fracaso parcial de la instrucción basada en computadora.

1.2.2. Dificultad de estimación de los recursos necesarios para la producción del software educativo

A medida que el costo del hardware para el desarrollo del software educativo disminuye, el gasto real de desarrollo de dicho software se ve incrementado por el riesgo al fracaso asociado a la producción del mismo. A pesar de todo, el uso de la computadora en la educación está creciendo, lo que impulsa la demanda de un software educativo de mejor calidad pero al mismo tiempo más accesible, por lo que, los costos de producción deben de ser bajos.

La aplicación de un enfoque tradicional en el desarrollo del software educativo, sustentado en la combinación de un diseño pedagógico, a menudo trae consigo altos costos y fechas imposibles de predecir. Por lo tanto, resulta conveniente explorar la posibilidad de aplicar un enfoque sistemático alternativo, el cual se oriente al control tanto de los costos como del calendario para el desarrollo del producto y en definitiva, a promover elevar la calidad [GJ02].

1.3. Planteamiento de una ingeniería del software educativo

Según Pressman [PR01] el software educativo, desde el principio, se propuso bajo una metodología de ingeniería de software que proporcionara al proceso de desarrollo una aproximación sistemática, basado en fundamentos y metodologías semejantes a las existentes en otras ingenierías tradicionales.

Una Ingeniería del Software Educativo ha de definir los principios, métodos y herramientas que soporten el desarrollo sistemático de calidad. Esto conlleva al desarrollo de modelos, procesos, metodologías y herramientas específicas para el software educativo, aprovechando sus características específicas frente a otro tipo de software. Además, se deben proporcionar procedimientos fiables para la estimación de recursos económicos, tecnológicos y humanos en el desarrollo del software educativo. De acuerdo a David

Martínez [MD02], la Ingeniería del Software Educativo se debe enfocar a la optimización del proceso de desarrollo mediante la máxima reutilización.

1.4. Ingeniería del Software: Una base para el desarrollo

El término software no sólo involucra programas, sino también todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. La ingeniería del software se define como “una disciplina que comprende todos los aspectos de la producción del software desde las etapas iniciales de especificación del sistema hasta el mantenimiento” [SI02].

La Ingeniería del Software es una tecnología multicapa. La Figura 1.2 muestra las capas que la constituyen. Cualquier enfoque de ingeniería, incluyendo la ingeniería del software, debe apoyarse sobre un compromiso de calidad [PR01].



Figura 1.2 Capas de la Ingeniería de Software [PR01].

De acuerdo a González [GH01] el objetivo de la Ingeniería del Software es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo se deben emplear métodos efectivos junto con herramientas modernas dentro del contexto de un proceso de desarrollo de software. Esto muestra la importancia de utilizar a la ingeniería del software para la construcción de un sistema de alta calidad y sin perder de vista el proceso que se sigue durante el desarrollo del mismo.

1.4.1. Proceso de Ingeniería del Software

Cuando se trabaja en la construcción de un producto, un sistema o se presta un servicio, es importante llevar a cabo una serie de pasos para lograr el objetivo planteado. Por tal motivo, es importante puntualizar qué es el proceso de ingeniería del software. Éste se define como “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad” [URL13].

El proceso de desarrollo del software requiere de un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama *ciclo de vida del software*. La construcción de un modelo de proceso ayuda a los desarrolladores a comprender la brecha que existe entre lo que debe ser y lo que es. Otras razones que indican la importancia de construir un modelo de proceso, son [LS02]:

La creación de un modelo de proceso ayuda al equipo de desarrollo o al desarrollador a encontrar las inconsistencias, las redundancias y las omisiones en el proceso. A medida que estos problemas se descubren y se corrigen, el proceso se vuelve más efectivo y concentrado en la construcción del producto final.

El modelo debe reflejar las metas del desarrollo, como la construcción de software de alta calidad, la localización temprana de los defectos en el desarrollo y el cumplimiento de las restricciones del cronograma.

Todo proceso se debe adaptar a la situación especial en que será utilizado.

Todo modelo de proceso del desarrollo de software incluye los requerimientos del sistema como entrada y la entrega de él como salida.

Existen muchos modelos del proceso de software, cada uno tiene sus propias ventajas y desventajas. Para determinar qué modelo es el adecuado en el desarrollo del software educativo de esta tesis, es necesario analizar los requerimientos del usuario entre otros factores.

En este trabajo de tesis se utiliza el Modelo de Cascada con Prototipado, la razón es que este modelo ayudó a conocer aspectos del sistema que no se pueden especificar durante la toma de requisitos y durante el diseño del sistema, dado que los juegos no se definen en una primera fase de análisis de requisitos, únicamente se presentaron bosquejos obtenidos de las necesidades de los usuarios. Durante la fase de diseño y con ayuda de los prototipos, estos bosquejos se redefinen con la continua intervención de los usuarios hasta el punto de obtener los juegos apropiados. La Figura 1.3 muestra dicho modelo.

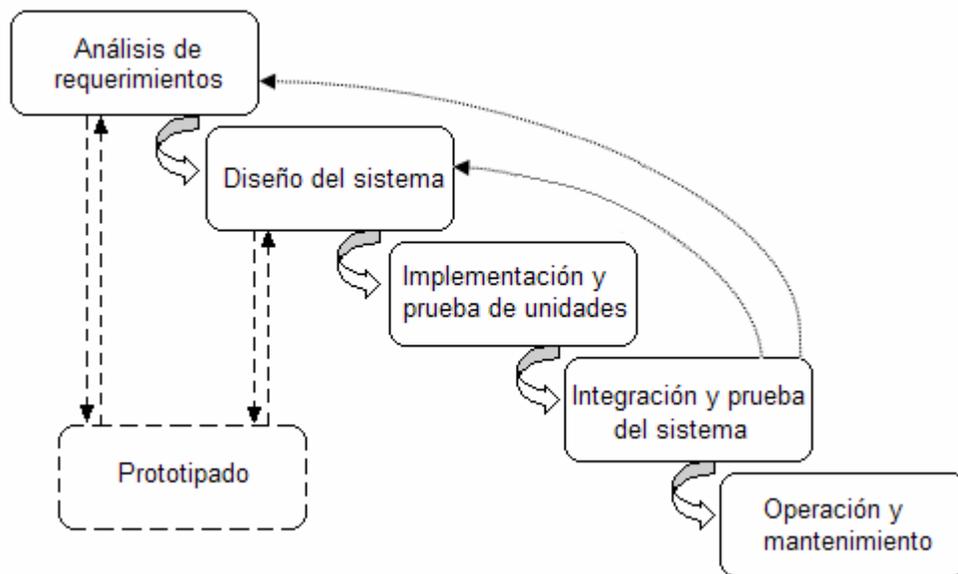


Figura 1.3 Modelo de cascada con prototipado [LS02].

El Modelo de Cascada también se le conoce como *Ciclo de Vida Básico*, fue propuesto por Winston Royce y ha sufrido varias modificaciones a lo largo del tiempo [PR01] [SI02]. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo las cuales son:

Análisis y definición de requerimientos: El proceso de reunión de requisitos se intensifica y se centra en el software especialmente. Para comprender la naturaleza del programa a construir, el Ingeniero de Software debe comprender el dominio de información del sistema, así como la función requerida, comportamiento, rendimiento e interconexión. Los servicios,

restricciones y metas del sistema se definen a partir de las consultas con los usuarios y sirven como especificación del sistema.

Diseño del sistema: El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos del programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental. El proceso del diseño traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación. Además, este proceso de diseño del sistema considera los requerimientos tanto de hardware como de software.

Implementación y prueba de unidades: Durante esta etapa, el diseño de software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una de ellas cumpla con su especificación.

Integración y prueba del sistema: Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software.

Operación y mantenimiento: El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del mismo una vez que se descubran nuevos requerimientos.

Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema y decidan si éste es adecuado para el producto terminado. El prototipado ayuda a los desarrolladores a evaluar las estrategias de diseño alternativas y decidir cuál es la mejor para un proyecto en particular.

La interfaz de usuario se construye y prueba como un prototipo, de modo que los usuarios comprendan qué aspecto tendrá el nuevo sistema y los diseñadores se orienten correctamente sobre la forma en que el usuario desea interactuar con el sistema. El prototipado es útil para asegurar que cada función trabaje correctamente y también asegura que los requerimientos se cumplan [LS02].

Capítulo 2

2. La usabilidad y su papel en la calidad del software

En el desarrollo del software se ha reconocido a la usabilidad como factor clave para el éxito de un producto de software. Cabe señalar que las técnicas de usabilidad que nos permiten alcanzar el nivel deseado en el software, pertenecen al campo de HCI (*Human Computer-Interaction*).

La usabilidad es una necesidad básica en el desarrollo del software, ya que ayuda a alcanzar niveles óptimos de calidad en el mismo. Robert Glass destaca la importancia de las GUI (Interfaz Gráfica de Usuario) [GR02] y Craig Larman sostiene que probablemente no hay otra técnica para el éxito del desarrollo de software como la ingeniería de usabilidad y el diseño de las interfaces de usuario [FX03].

2.1. Usabilidad

Nielsen define a la *usabilidad* como “medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción de un contexto específico” [NJ95].

Este término se asocia a cinco atributos que deben ser tratados en el desarrollo del software [NJ93]:

Facilidad de aprendizaje: El sistema deberá ser fácil de aprender, de forma que el usuario pueda trabajar rápidamente con él.

Eficiencia: Determina el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la velocidad máxima de tareas realizadas por el usuario.

Memoria: Los usuarios que no utilizan el sistema regularmente, deben ser capaces de usarlo sin tener que aprender cómo funciona cada vez que lo utilicen.

Errores: Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario. Errores fatales no deben ocurrir en el sistema.

Satisfacción: Este es el atributo más subjetivo, ya que muestra la impresión que el usuario tiene del sistema. Los usuarios deben sentirse a gusto con el uso del mismo.

Ferré menciona que algún atributo puede que esté más marcado que otro dentro de un sistema, pero esto no significa que la usabilidad del mismo sea incorrecta [FX00].

2.1.1. La importancia de la usabilidad

La primera razón por la cual se aplica la usabilidad en el desarrollo de un software, es la obtención de un sistema que hace al usuario más productivo, aumentando su eficiencia

y satisfacción al utilizarlo [FX00]. Si el sistema no es percibido como una herramienta que ayude al usuario a realizar sus tareas, se dificulta la aceptación del mismo. Si las tareas del usuario no se respaldan convenientemente por el sistema, entonces no se está respondiendo de manera adecuada a las necesidades del mismo.

Los sistemas construidos con criterios de usabilidad aventajan a aquellos que no los utilizan, entre las ventajas se pueden mencionar [LM02]:

Mejora en la productividad

Reducción del costo y del tiempo de aprendizaje

Incremento de la autonomía de los usuarios finales

Lauder [LT95] observó que el 80% de los costos de mantenimiento se deben a problemas de los usuarios con el sistema y no a problemas técnicos. Además, indica que el 64% de estos costos están relacionados con problemas de usabilidad. Estudios sobre el tema de la productividad en herramientas software concluyen que la facilidad de uso y aprendizaje son las razones más importantes a la hora de incrementar de forma efectiva la productividad de los usuarios. Estas situaciones resaltan la importancia de la usabilidad y justifican la necesidad de incorporarla antes, durante y después del desarrollo del software.

Esto implica que el desarrollo de aplicaciones es una actividad multidisciplinaria que incorpora factores humanos, técnicas y conocimientos ergonómicos con el objeto de mejorar las condiciones de trabajo del usuario.

2.1.2. Interacción Humano-Computadora (HCI)

Cuando las personas y la computadora interactúan lo hacen a través de un medio o interfaz. Una interfaz es una superficie de contacto que refleja las propiedades físicas de los que interactúan, las funciones a realizar, el balance de poder y control [LJ01].

2. La usabilidad y su papel en la calidad del software

La *Interacción Humano-Computadora* es una ciencia multidisciplinaria, donde sus conocimientos comprenden: computación, psicología, ergonomía, tecnologías de información, diseño gráfico, sociología, etc., el término a alcanzar es la usabilidad [SB98].

La Figura 2.1 muestra gráficamente todos los aspectos relacionados con la Interacción Humano-Computadora. Además, deja entrever la importancia de implicar personas procedentes de diferentes disciplinas y áreas de conocimiento.

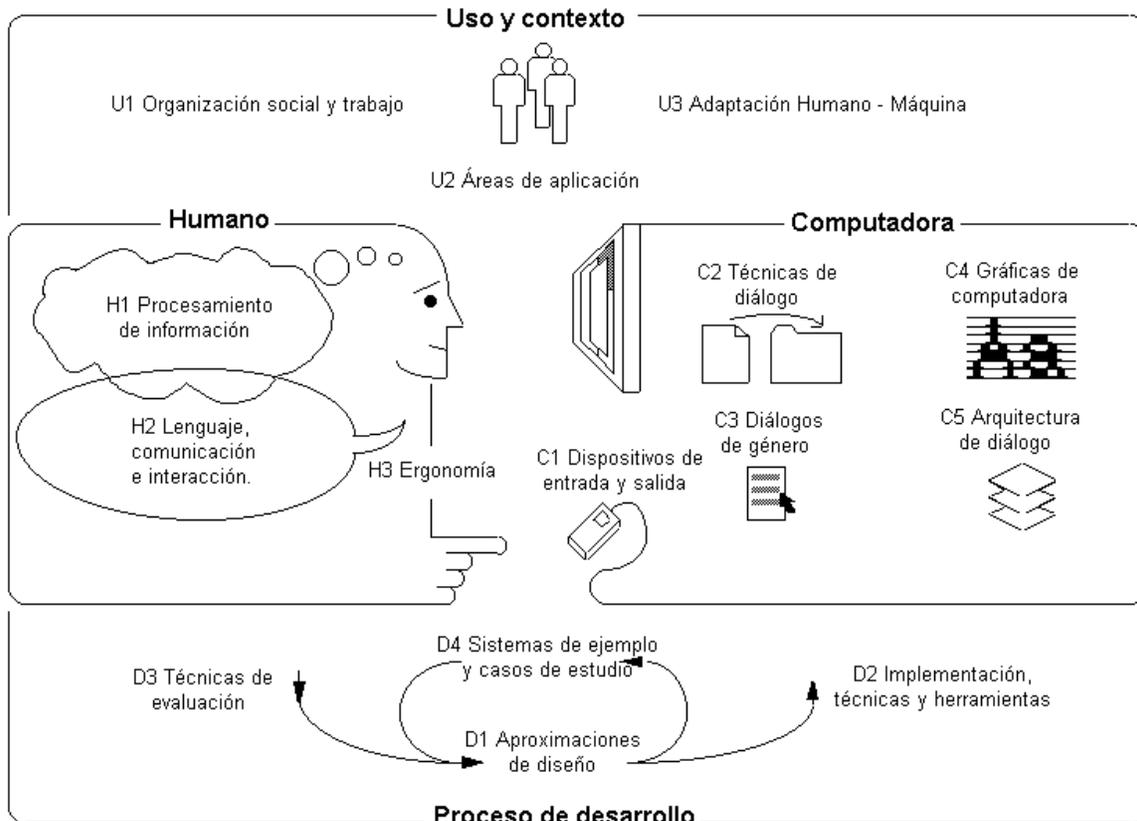


Figura 2.1. Esquema de HCI [URL05]

Además, el esquema refleja la idea de que el individuo no está solo, sino que realiza su trabajo dentro de una determinada organización social. Para que ello sea posible existe un proceso complejo de desarrollo en el que cada uno de sus componentes debe ser abordado con igual grado de implicación y no caer en el error de evadir la parte humana y centrarse solamente en la parte tecnológica [GT04].

HCI es importante en el proceso de diseño ya que la interfaz es el primer punto de contacto del usuario con el sistema y el usuario hará un juicio de éste en base a la interfaz presentada [FC97].

2.1.3. Diseño Centrado en el Usuario (UCD)

Los sistemas son utilizados por usuarios, por lo que no debemos olvidar y aislar a la fase final de un proyecto a los usuarios. El diseño de sistemas implica realizar un diseño donde el usuario sea el centro de atención y la implementación de las funciones del sistema se realice de acuerdo a las características del mismo.

El punto que no se debe omitir es que centrarse en el usuario significa centrarse en todos los usuarios, sin que ello indique que se deba incorporar a todos los posibles usuarios de un determinado sistema, si no que se deben contemplar todos los rasgos y diferencias entre ellos [SC95].

El concepto de *Diseño Centrado en el Usuario (UCD)* se define como "un método que hace directamente la pregunta central de ¿cómo es para el usuario, la experiencia del uso del software?"[ND86].

En esta metodología es importante involucrar al usuario final en el desarrollo del sistema. De esta forma, se puede mejorar la utilidad (relevancia) y usabilidad (facilidad de uso) de cualquier pieza de software [TA98]. El diseño centrado en el usuario tiene como objeto hacer la vida del usuario más sencilla a la hora de trabajar con un sistema.

Shneiderman [SB98] propone los puntos que deben considerarse para desarrollar un software de calidad de acuerdo a la metodología de UCD son: funcionalidad, fiabilidad, disponibilidad, seguridad, integridad de datos, estandarización, integración, consistencia, portabilidad, ajustes en calendarios y presupuestos.

La metodología UCD descrita por Jenny Preece [PJ93] es de gran importancia para el desarrollo del software, la cual consta de los siguientes pasos:

Análisis de requerimientos o definición de mercado: Identifica a los usuarios a quienes el producto va dirigido, identifica competidores, y determina las necesidades básicas que deberán cumplirse para que el producto tenga éxito.

Análisis de tareas: Identifica y entiende las metas y tareas de los usuarios, las estrategias que utilizan para realizar su trabajo, las herramientas que actualmente usan, cualquier problema que experimenten y los cambios que les gustaría ver en sus tareas y herramientas.

Diseño: Utilizando los resultados del análisis de tareas, se crea una propuesta de solución alternativa, se solicita retroalimentación a través de sesiones de diseño “cognitive walk-through”⁴ con los usuarios, y se elige una solución basada en la ayuda de los mismos.

Implementación: Se refiere al uso de herramientas de implementación, como son: el lenguaje de programación y el hardware que se requiere, entre otros.

Pruebas de usabilidad: Se solicita la retroalimentación periódicamente de los usuarios sobre el sistema en desarrollo.

Esta metodología es la utilizada en el desarrollo de esta tesis. La Figura 2.2 ilustra el proceso iterativo de la metodología UCD.

⁴ Se conoce en Español como “paseo cognitivo”.

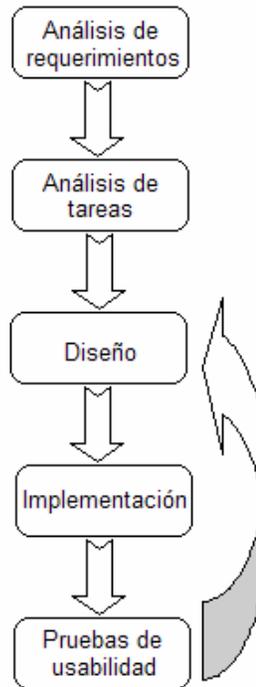


Figura 2.2. Proceso iterativo de la Metodología Centrada en el Usuario [PJ93].

Existe una relación entre el Modelo de Cascada con Prototipado y la metodología UCD, ya que el proceso que se sigue es parecido y se complementan para poder obtener los resultados deseados a las necesidades del software educativo.

2.1.4. Métodos utilizados en pruebas de usabilidad

Existen varios métodos para evaluar y probar la usabilidad de un software. Algunas diferencias entre estos radican en el número de usuarios que se consideran para realizar las pruebas y la etapa de desarrollo del software en que se pueden aplicar, siendo posible combinar varios métodos como son:

Protocolo de pensamiento en voz alta (*Think Aloud Protocol*): Cuando el usuario está realizando alguna tarea, se le solicita expresar en voz alta sus pensamientos y opiniones mientras interactúa con el producto, con la finalidad de que los desarrolladores entiendan cómo visualiza el usuario el software y así poder identificar errores. Su principal ventaja radica en que

es una prueba económica, ya que sólo necesita de tres a cinco usuarios, además de que permite identificar y entender las ideas que se tienen acerca del sistema. Por el contrario, su desventaja radica en que es difícil para los usuarios expresar lo que piensan [NJ93].

Paseo cognitivo (*Cognitive Walk-Through*): Es una técnica de revisión en la que evaluadores expertos construyen escenarios de las tareas de una especificación o de un prototipo inicial, donde los evaluadores expertos trabajan con la interfaz, como si fueran un usuario más, es decir, *realizan un recorrido por la interfaz*. A cada paso el usuario revisa la interfaz, si encuentra problemas, indica que existe un error en la misma. Se sugiere ir evaluando las especificaciones que se vayan teniendo del sistema. [URL12].

Evaluación heurística: Es una técnica aplicada por usuarios expertos, la cual consiste en verificar el sistema frente a un conjunto pequeño de heurísticas de diseño [URL01].

Entrevistas: A través de las entrevistas (por teléfono o en persona), se obtienen descripciones de las reacciones y dificultades encontradas en el sistema [URL02].

Observaciones: Se aplica esta técnica para saber qué hacen los usuarios cuando están interactuando con el sistema. Existen dos tipos de observaciones: formales e informales. Las observaciones formales utilizan recursos tecnológicos como la grabación de video. Las informales consisten en que, la persona que observa realiza anotaciones sobre lo que está viendo [URL02].

En el capítulo siguiente se describen qué métodos fueron utilizados para las pruebas de usabilidad de este trabajo de tesis.

Capítulo 3

3. Análisis y Diseño del Software Educativo

En este capítulo se presenta a detalle la forma en que se realizó el análisis y diseño del software educativo planteado. Además, se mencionan las características del modelo del proceso de enseñanza que son de utilidad para el proceso de diseño.

3.1. Análisis de requisitos del software educativo EZ-KET

En esta fase se parte de los objetivos que se obtuvieron al inicio del proyecto. Éstos se tienen que refinar para poder trabajar con ellos, además de tomar en cuenta las necesidades de profesores y estudiantes para realizar el proceso de diseño del sistema.

Un punto importante es que el término *requisitos* a veces se utiliza de forma inconsistente, ya que puede designar tanto una descripción abstracta de alto nivel de lo que el usuario espera del sistema, así como también una descripción más detallada y formal del mismo [DA93] [SI97].

Para poder refinar los objetivos principales se tuvo que platicar con los profesores y los estudiantes. Con los estudiantes se realizaron pruebas de usabilidad para determinar parte de los requerimientos del sistema. Estas pruebas consistieron en mostrar prototipos en papel de juegos en nivel KET a los estudiantes, los cuales realizaron comentarios acerca del mismo: qué opinaban del juego, si les gustaría verlo implementado en computadora, si creían necesario incluir alguna descripción a algún elemento, entre otros. Todos los comentarios fueron aceptados y se analizó la forma de incluirlos en los nuevos prototipos. A continuación se presentan los primeros prototipos realizados en papel y las opiniones generales de los usuarios acerca de cada prototipo.

3.1.1. Prototipo: Ahorcado

La Figura 3.1 muestra el primer prototipo de juego del ahorcado. Este prototipo fue aceptado por todos los estudiantes, se realizaron las pruebas de usabilidad con ocho alumnos de nivel KET. En general, éstos se mostraron complacidos con el juego, la forma en que es representado y la distribución de los elementos.



Figura 3.1. Primer prototipo del juego del ahorcado

3.1.2. Prototipo: Cachar palabras

El juego de cachar palabras (Figura 3.2), a diferencia del juego del ahorcado, si recibió algunos comentarios importantes:

Los usuarios se mostraron inquietos acerca de la velocidad con que cada palabra bajaría por la ventana, sugirieron una velocidad moderada en la caída de cada una de las mismas.

Le faltaba color al fondo de la ventana. El nuevo prototipo debía contener un color diferente al blanco.

Los usuarios preguntaron si en lugar de caer palabras podrían caer frases.

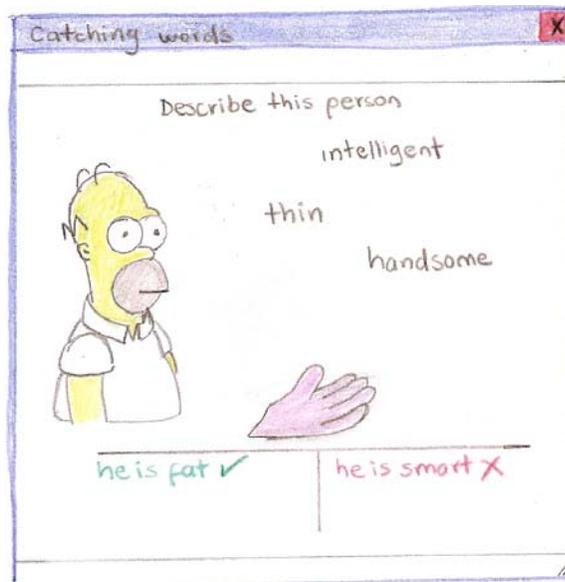


Figura 3.2. Primer prototipo del juego de cachar palabras

3.1.3. Prototipo: Mapa

El juego del mapa que se presenta en la Figura 3.3 no sufrió cambios considerables en esta primera fase debido a los pocos comentarios de los usuarios en esta prueba. Las aportaciones más importantes que los usuarios hicieron fueron:

El mapa debería ser más grande y ocupar toda la pantalla. Sugirieron que la ventana del mapa apareciera maximizada.

Contar con una mayor cantidad de jugares a visitar en el mapa.

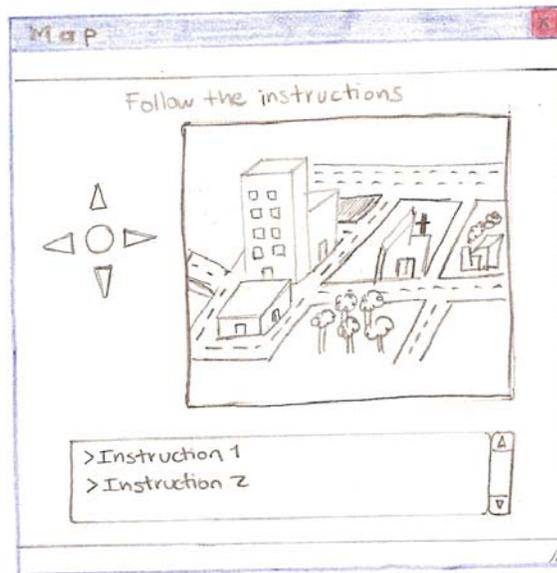


Figura 3.3. Primer prototipo del juego del mapa

3.1.4. Prototipo: Relacionar imágenes con palabras

El juego de relacionar palabras con imágenes (Figura 3.4), no recibió comentarios negativos, ya que a los estudiantes les agradó la forma en que era presentado el juego y todos sus elementos.

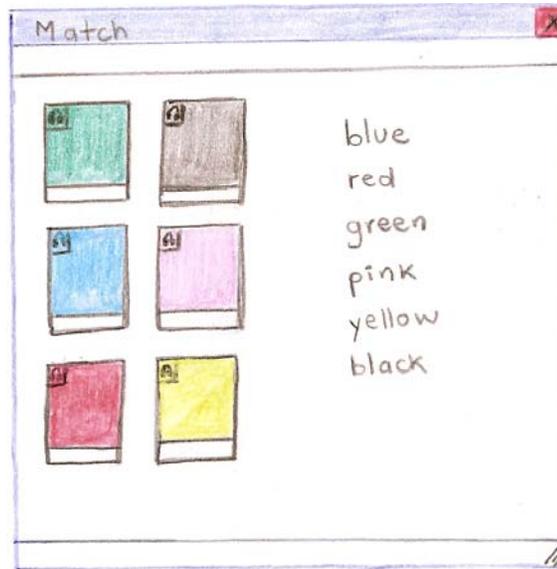


Figura 3.4. Primer prototipo del juego de relacionar palabras con imágenes

3.1.5. Prototipo: Jugando con los verbos

En cuanto al juego de verbos (Figura 3.5), los estudiantes se mostraron inquietos acerca del número de casillas que deberían abrir para terminar el juego. Sugirieron que al completar una cantidad de puntos el juego se diera por terminado sin necesidad de tener que abrir todas las casillas del tablero. Además, sugirieron un cambio en los colores del mismo.

Verbs Game				
★ 200				
Verbs starting with C	Verbs starting with H	Verbs starting with K&L	Verbs starting with P, Q and R	Verbs starting with W
200	200	200	200	200
400	200	400	400	200
600	400	600	400	400
800	600	800	400	800
1000	600	1000	600	800

Figura 3.5. Primer prototipo del juego de verbos

Todas las opiniones y sugerencias acerca de cada uno de los prototipos fueron de utilidad para determinar la forma en que se desarrollaría el sistema y aclarar los requisitos del mismo.

Una vez incorporados los comentarios de los estudiantes a los prototipos, éstos se mostraron únicamente a los profesores del área de Inglés, ya que la intención era definir el contenido de cada juego y ellos tendrían una opinión más clara sobre qué conceptos resaltar y cuáles omitir en cada uno de los juegos. La información obtenida fue de utilidad tanto para ayudar a definir la interfaz, como para determinar las características de cada juego con respecto a los niveles KET. Los requisitos de cada juego se muestran en la Tabla 3.1.

Tabla 3.1. Requisitos identificados en cada juego.

Juegos	KET 1	KET 2	KET 3
<i>Ahorcado</i>	Este juego tiene una imagen de la palabra a adivinar e incluye la pronunciación de la misma.	Tiene una pequeña descripción y cuenta con audio de la palabra a adivinar.	Solamente cuenta con la pronunciación de la palabra a adivinar.
<i>Relacionar imágenes con palabras</i>	El juego consta de imágenes y palabras, donde cada palabra se debe arrastrar a la imagen correspondiente. Se incluye audio para ejercitar la habilidad de “listening”. Las palabras incluidas en este juego dependen del nivel de KET.		
<i>Cachar palabras</i>	Consta de una imagen que tiene asociada frases correctas e incorrectas. Las frases caen y deben ser atrapadas por medio de un carrito de supermercado. La velocidad en la que caen las palabras depende del nivel de KET.		
<i>Mapa</i>	El juego está formado por una imagen de una ciudad. Las instrucciones aparecen en pantalla, se incorpora el sonido de la instrucción.	El juego está formado por una imagen de una ciudad, únicamente se presenta una grabación de la instrucción.	
	La complejidad depende del nivel de KET.		
<i>Ejercicio tipo A</i>	Este ejercicio está formado por una pregunta o la descripción de algún tema en particular, donde se presentan cuatro imágenes como posibles respuestas, pero sólo una es la correcta. La complejidad de cada pregunta depende del nivel de KET.		
<i>Ejercicio tipo B</i>	Consta de una pregunta o la descripción de algún tema en particular, donde se presentan cuatro opciones como posibles respuestas y sólo una es la respuesta correcta. La complejidad de cada pregunta o descripción depende del nivel de KET.		
<i>Jugando con los verbos</i>	Este juego no está disponible para este nivel de KET, ya el temario no contempla verbos en pasado.	Este juego sirve para practicar los verbos en pasado. Consta de un tablero, el cual contiene una serie de verbos, donde cada uno tiene un puntaje asociado.	
<i>Casa</i>	El juego requiere que el estudiante identifique las partes de una casa. Incluye audio para practicar la habilidad de “listening”.	Cada parte de la casa muestra los muebles representativos del lugar, así como, las palabras de cada mueble, las cuales se deben arrastrar a la imagen correspondiente.	El juego se divide de acuerdo a las partes de una casa. Cada parte muestra los muebles representativos del lugar, se debe escribir la palabra correspondiente a cada mueble.

El temario correspondiente a cada nivel KET no se especificó por completo al momento de realizar el análisis, ya que dicho temario está sujeto a cambios en los cursos de la Universidad Tecnológica de la Mixteca. Dada la flexibilidad del temario, el sistema se diseñó para soportar la inserción de nuevos temas que serían agregados por los profesores una vez entregado el sistema. El temario parcial utilizado para el análisis del sistema se presenta en el Anexo A.

3.2. Pruebas de usabilidad

Las pruebas de usabilidad generan retroalimentación al diseño del sistema, permitiendo que el software satisfaga las necesidades de los usuarios (alumnos y profesores). Para realizar dichas pruebas se requiere de un espacio, donde se pueda tener a los usuarios, equipo de cómputo y el sistema a evaluar. Para el desarrollo de las pruebas de usabilidad de este trabajo de tesis, se utilizó el Laboratorio de Usabilidad (UsaLab)⁵ de la Universidad Tecnológica de la Mixteca (UTM).

Como se mencionó en el capítulo anterior, existen varios tipos de pruebas de usabilidad. Las que se utilizaron en esta tesis fueron: Protocolo de Pensamiento en Voz Alta (realizadas únicamente con estudiantes de Inglés de la UTM) y Paseo Cognitivo (realizadas por profesores de Inglés de la UTM).

Se realizaron pruebas de usabilidad a seis estudiantes de diferentes niveles KET. La Figura 3.6 muestra un estudiante realizando las pruebas.

⁵ <http://mixtli.utm.mx/~usalab>



Figura 3.6. Prueba de usabilidad en curso.

Durante las pruebas de usabilidad los estudiantes realizaron tareas como: registrar usuario, recordar contraseña, ingresar al juego y utilizarlo.

3.2.1. Resultado de las pruebas de usabilidad

A los estudiantes se les plantearon las siguientes preguntas:

¿Esta tarea fue fácil de hacer?

- ¿Qué tan satisfecho estuviste con la tarea?

¿Tuviste problemas con la tarea?

¿Tienes alguna propuesta para mejorar el sistema?

Tarea: Registrar usuario

Los resultados obtenidos en la tarea de *registro*, fue la siguiente:

¿Esta tarea fue fácil de hacer? Figura 3.7.

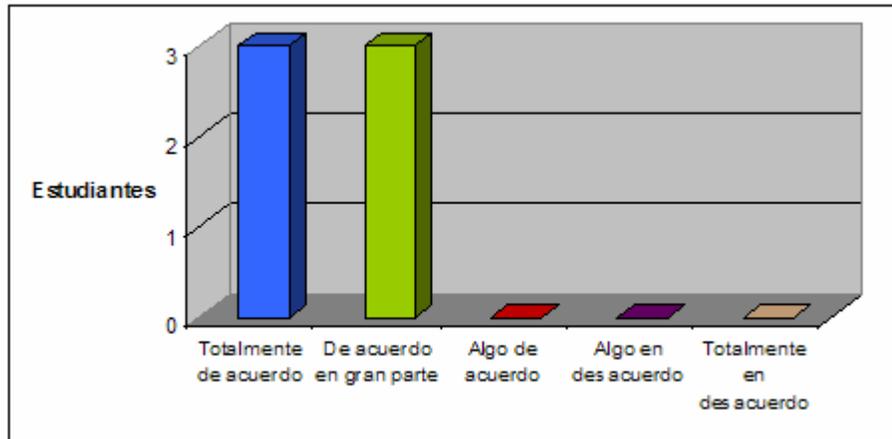


Figura 3.7. Gráfica comparativa de la tarea registrar usuario.

¿Qué tan satisfecho estuviste en la tarea? ¿Qué cara corresponde a tu estado de ánimo? Figura 3.8.

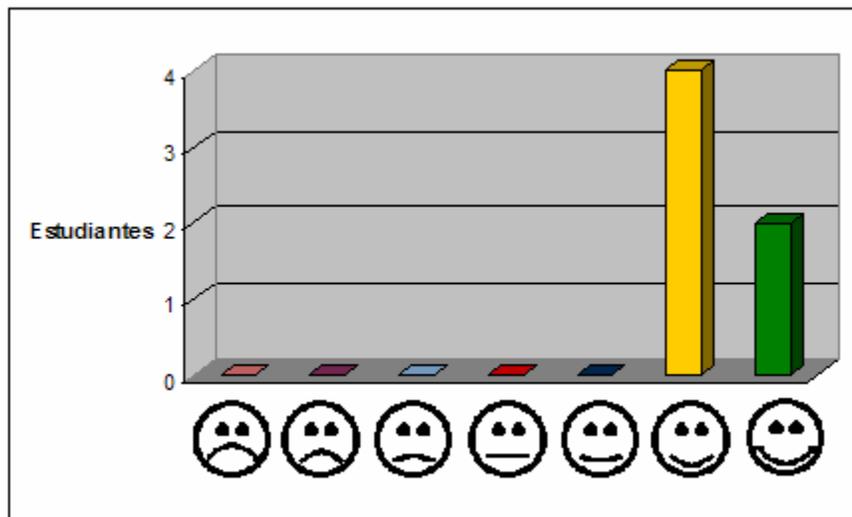


Figura 3.8. Gráfica correspondiente a la tarea registrar usuario.

¿Tuviste problemas con la tarea?

Ningún estudiante tuvo problemas con esta tarea

¿Tienes alguna propuesta para mejorar el sistema?

Propusieron que la ayuda estuviera en español.

Tarea 2: Recordar contraseña

Los resultados obtenidos en la tarea de *recordar contraseña*, fue la siguiente:

¿Esta tarea fue fácil de hacer? Figura 3.9.

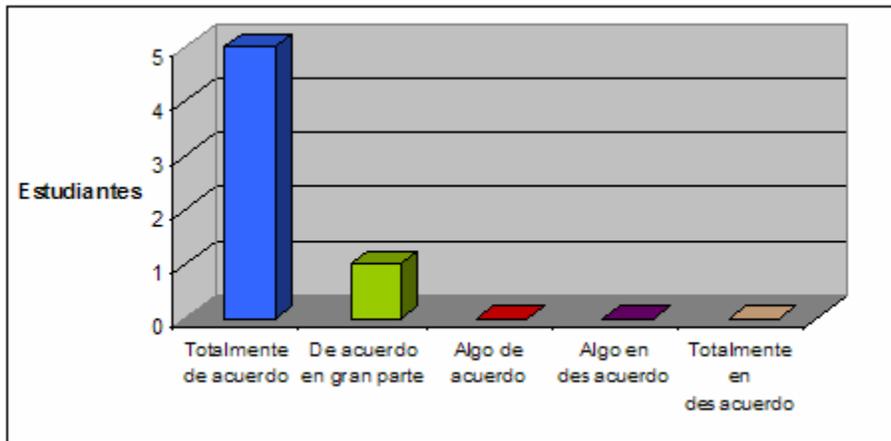


Figura 3.9. Gráfica comparativa de la tarea recordar contraseña.

¿Qué tan satisfecho estuviste en la tarea? ¿Qué cara corresponde a tu estado de ánimo? Figura 3.10.

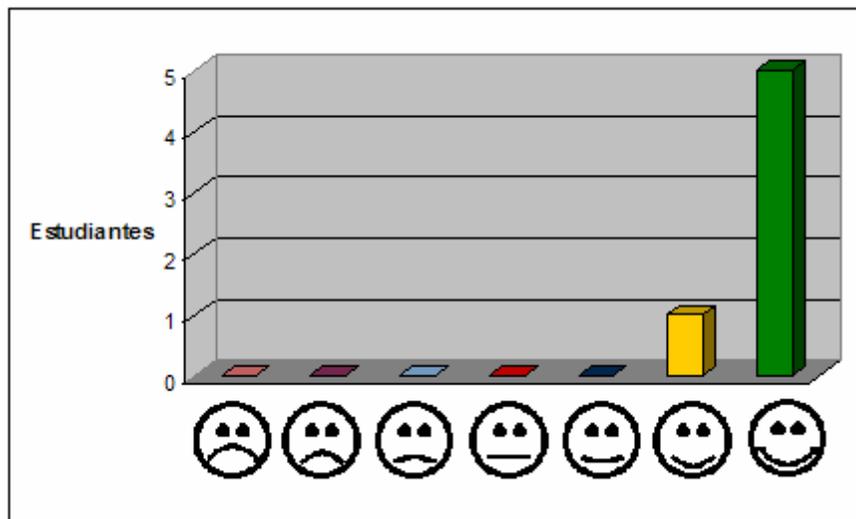


Figura 3.10. Gráfica correspondiente a la tarea recordar contraseña.

¿Tuviste problemas con la tarea?

Ningún estudiante tuvo problemas con esta tarea

¿Tienes alguna propuesta para mejorar el sistema?

Para esta tarea no hubo sugerencias, ya que se les hizo muy fácil.

Tarea 3: Entrar al juego y utilizarlo

Los resultados obtenidos en la tarea de *entrar al juego y utilizarlo*, fue la siguiente:

¿Esta tarea fue fácil de hacer? Figura 3.11.

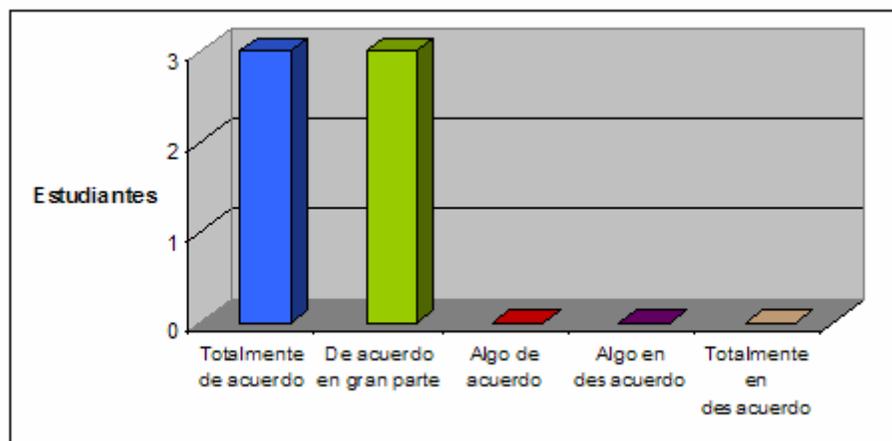


Figura 3.11. Gráfica comparativa de la tarea entrar al juego y utilizarlo.

¿Qué tan satisfecho estuviste en la tarea? ¿Qué cara corresponde a tu estado de ánimo? Figura 3.12.

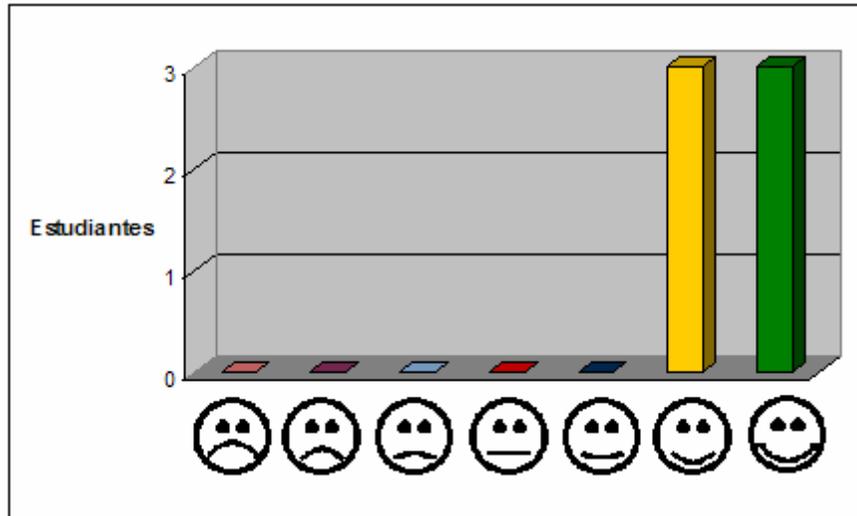


Figura 3.12. Gráfica correspondiente a la tarea entrar al juego y utilizarlo.

¿Tuviste problemas con la tarea?

Ningún estudiante tuvo problemas con esta tarea

¿Tienes alguna propuesta para mejorar el sistema?

Algunos estudiantes propusieron que la ayuda fuera en Inglés pero con términos sencillos. Otros sugirieron que la ayuda estuviera en Español.

3.3. Análisis de requisitos del sistema de retroalimentación

El sistema de retroalimentación que ayuda a los profesores a anexar temas en cada nivel KET, también se realizó bajo un primer prototipo (Figura 3.13 y Figura 3.14). Este prototipo proporcionó información útil al momento de definir la interfaz y la forma en que los datos se presentarían. Los términos utilizados para las instrucciones o etiquetas de cada ventana fueron propuestos por los profesores.

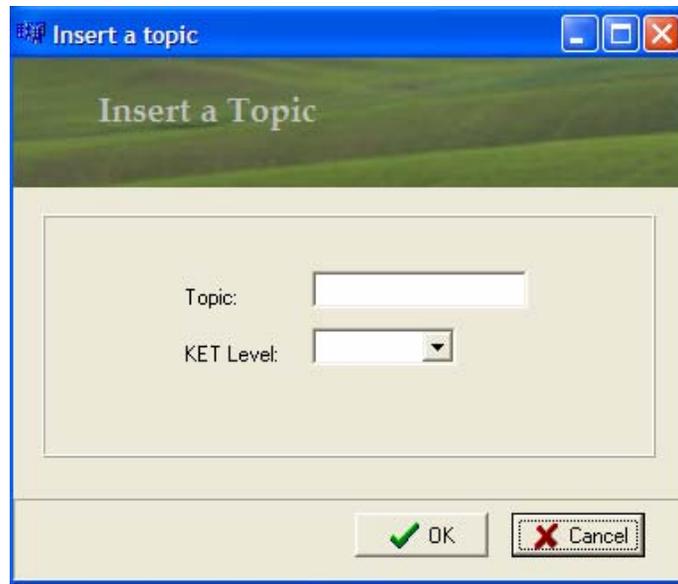


Figura 3.13. Primer prototipo para agregar un tema en el sistema de retroalimentación.

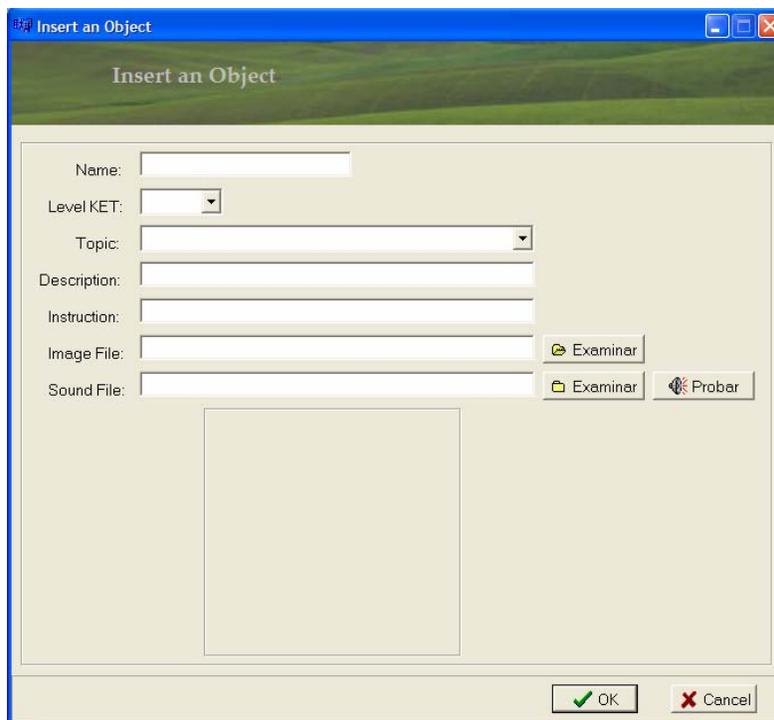


Figura 3.14. Primer prototipo del sistema de retroalimentación.

3.3.1. Refinamiento de requisitos a través de pruebas de usabilidad en el sistema de retroalimentación

Se realizaron pruebas de usabilidad a cuatro profesores del Centro de Idiomas de la UTM. En las pruebas de Paseo Cognitivo, los profesores aportaron las siguientes opiniones y sugerencias:

En el módulo donde se introduce la información correspondiente a las preguntas de verbos, sugirieron que se cambiara el nombre que aparece en el menú, ya que tenía el nombre de “*Questions*” y se pidió cambiar por “*Verbs Questions*”.

Sugirieron que en la ayuda aparezca un ejemplo de cómo introducir la información en cada módulo. También solicitaron se especificara en la ayuda el formato de las imágenes que soporta la aplicación.

Otra sugerencia fue que se cambiara el color del encabezado de las ventanas.

Una vez realizadas estas pruebas, se realizaron los cambios sugeridos.

3.4. Diseño del Software

El diseño del software describe la estructura del sistema a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y los algoritmos utilizados [SI02]. La fase de diseño identifica la forma más conveniente de implementar un sistema que pueda cubrir los requerimientos determinados en la fase de análisis.

3.4.1. Diseño del software educativo EZ-KET

A partir de las pruebas de usabilidad, se pudo llevar a cabo el diseño de la interacción usuario–sistema a través de los casos de uso.

En la Figura 3.15 se observa el diagrama de casos de uso del software educativo desarrollado en esta tesis.

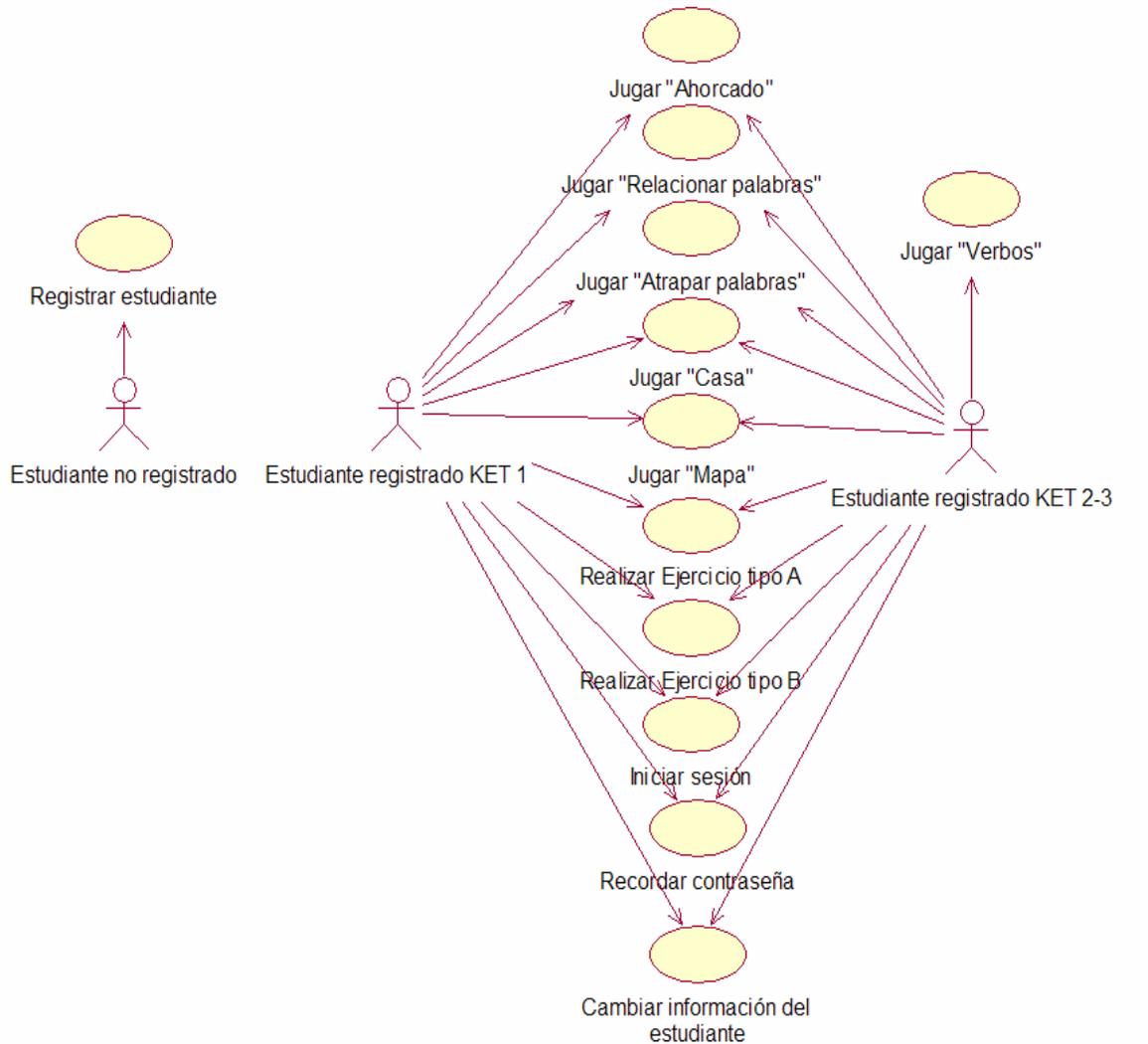


Figura 3.15. Diagrama de casos de uso del software educativo de alumnos.

Con la información obtenida de los casos de uso fue posible definir las respuestas del sistema esperadas por los usuarios, lo cuál facilitó la generación de los nuevos prototipos y simplificó el trabajo posterior en relación al comportamiento del sistema. A continuación se presenta un caso de uso para visualizar mejor la interacción del sistema con el usuario. Para una descripción más detallada acerca de los casos de uso identificados consulte el archivo ezket.pdf que se incluye en el CD.

Especificación de Caso de Uso: Jugar ahorcado

1. Jugar ahorcado

1.1 Breve descripción

Este caso de uso describe de forma general la secuencia de pasos que debe realizar un usuario para *jugar ahorcado*.

2. Flujo de eventos

Actores identificados

El único actor identificado en este caso de uso es el usuario registrado.

Tipo del Caso de Uso

Este caso de uso es primario, ya que de él depende que el usuario pueda hacer uso de la funcionalidad completa del sistema.

Flujo básico

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el usuario registrado accede a la pantalla de <i>jugar ahorcado</i> .	2. El sistema muestra en la parte inferior de la pantalla las letras del abecedario, del lado izquierdo de la pantalla se muestran los errores, en la parte central aparecen líneas que corresponden a la palabra a adivinar. En el lado derecho de la pantalla aparece el botón de audio para poder escuchar la palabra a adivinar. Si el juego es para un estudiante de KET 2 aparece una descripción de dicha palabra.
3. Presiona las letras del abecedario para construir la palabra o el botón de audio para escuchar dicha palabra.	4. Verifica que las letras seleccionadas correspondan a las de la palabra a adivinar y el sistema guarda la información (E1).

2.4 Flujos alternativos

E1. Se proporcionan letras que no corresponden a la palabra a adivinar y dicha información se guarda en el sistema.

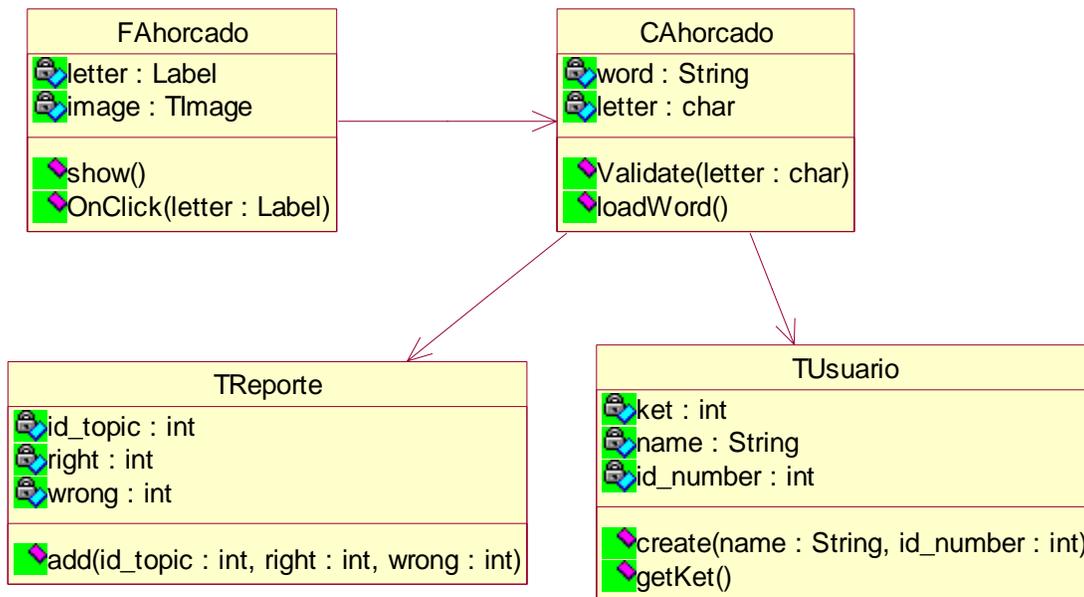
3. Requerimientos especiales

Ser un usuario registrado en el sistema y contar con un nivel de KET.

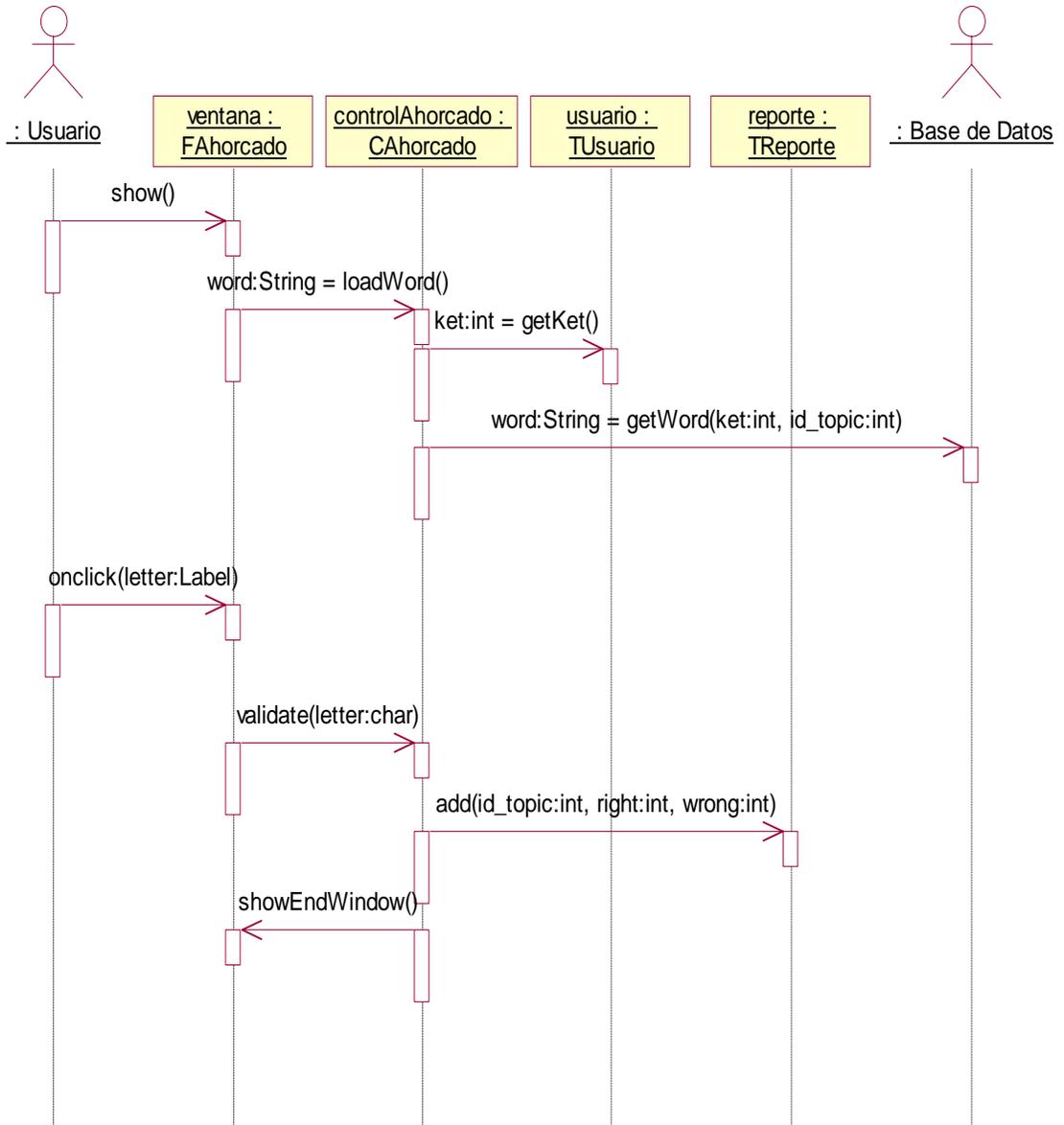
4. **Poscondiciones**

Al terminar este caso de uso, el usuario es capaz de hacer uso de la funcionalidad completa del sistema.

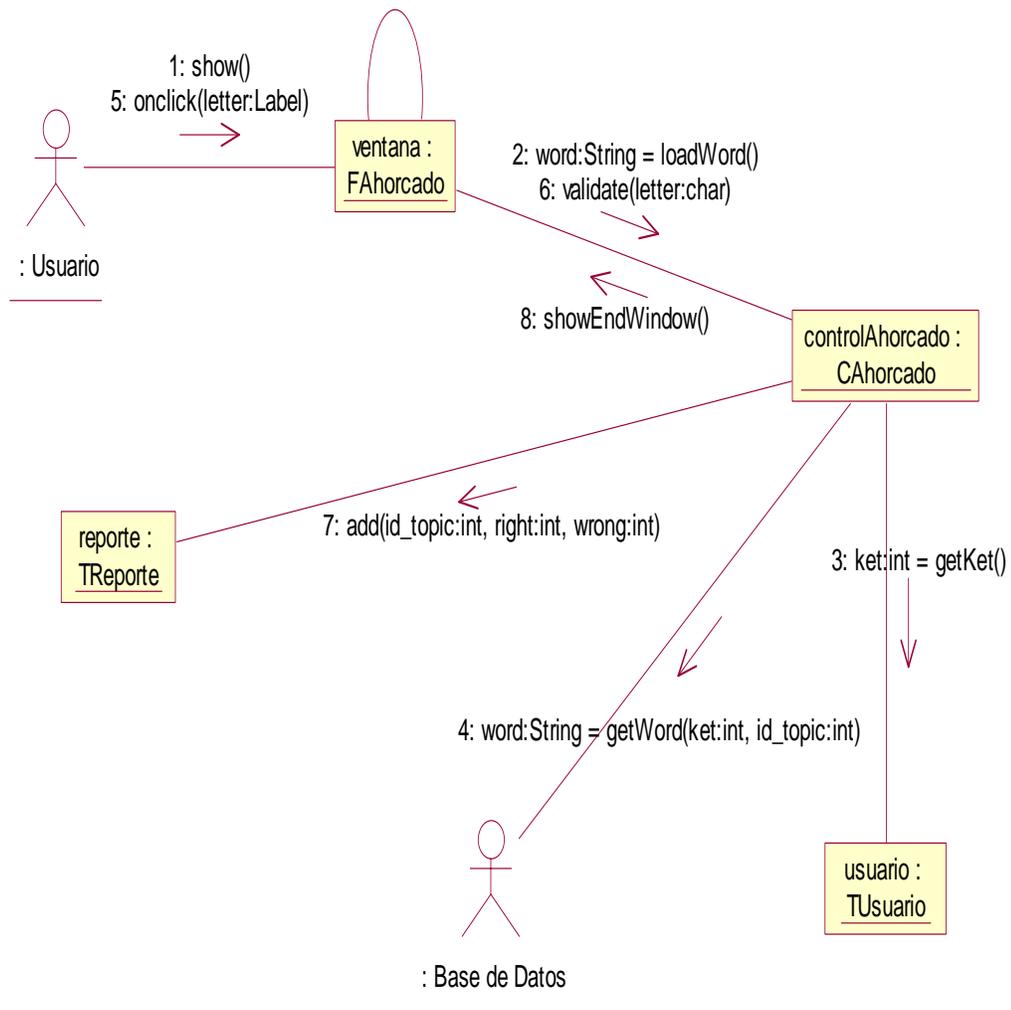
5. **Diagrama de clases**



6. Diagrama de secuencia del caso de uso *jugar ahorcado*: Flujo Básico



7. Diagrama de colaboración



3.4.2. Diseño del sistema de retroalimentación

El diagrama de casos de uso obtenido para la retroalimentación del sistema se muestra en la Figura 3.16.

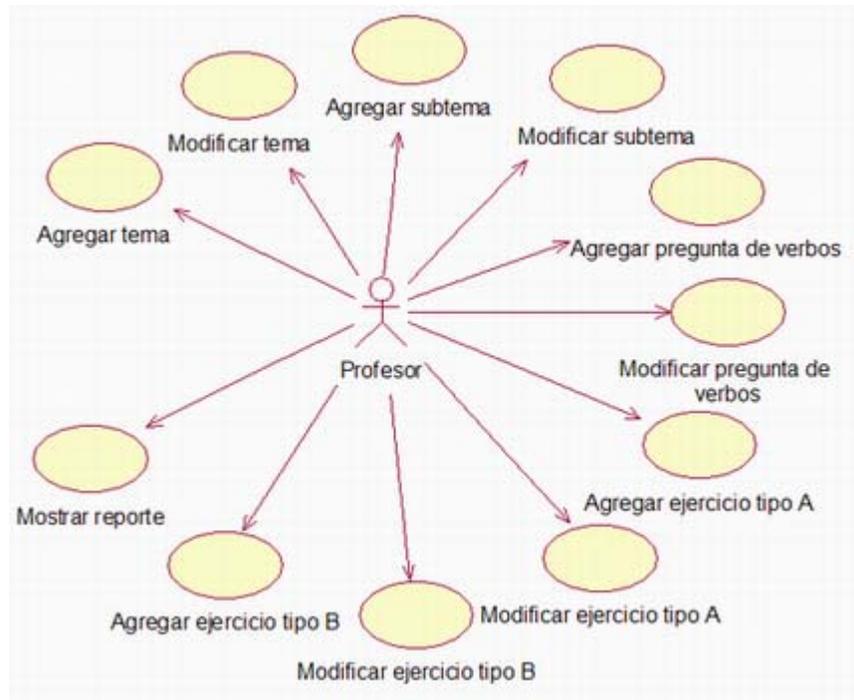


Figura 3.16. Diagrama de casos de uso del sistema de retroalimentación.

Para una descripción más detallada acerca de los casos de uso identificados consulte el archivo retroalimentación.pdf que se incluye en el CD. A continuación se presenta un caso de uso.

Especificación de Caso de Uso: Agregar subtema

1. Agregar subtema

1.1 Breve descripción

Este caso de uso describe de forma general la secuencia de pasos que debe realizar un usuario para *Agregar subtema*.

2. Flujo de eventos

Actores identificados

El único actor identificado en este caso de uso es el profesor.

Tipo del Caso de Uso

Este caso de uso es primario, ya que de él depende que el profesor pueda hacer uso de la funcionalidad completa del sistema.

Flujo básico

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando el profesor accede a la pantalla de <i>Agregar subtema</i> .	2. El sistema muestra en pantalla la opción para seleccionar un nivel de KET, seleccionar un tema, escribir el nombre del subtema, entre otra información solicitada. También se presentan los botones OK y CANCEL.
3. Selecciona un nivel de KET, selecciona un tema y proporciona toda la información solicitada en pantalla. Presiona el botón OK (E1).	4. Verifica que los valores ingresados en los campos sean válidos (E2). El sistema almacena la información introducida.

2.4 Flujos alternativos

- E1. Se presiona el botón CANCEL. La información no es almacenada en el sistema.
- E2. La información proporcionada es incorrecta.

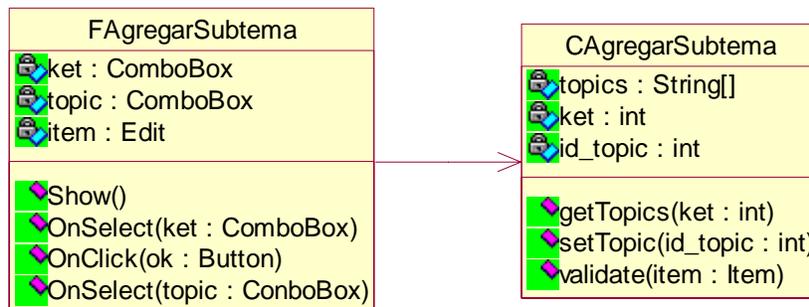
3. Requerimientos especiales

Ser profesor del centro de idiomas de la Universidad Tecnológica de la Mixteca.

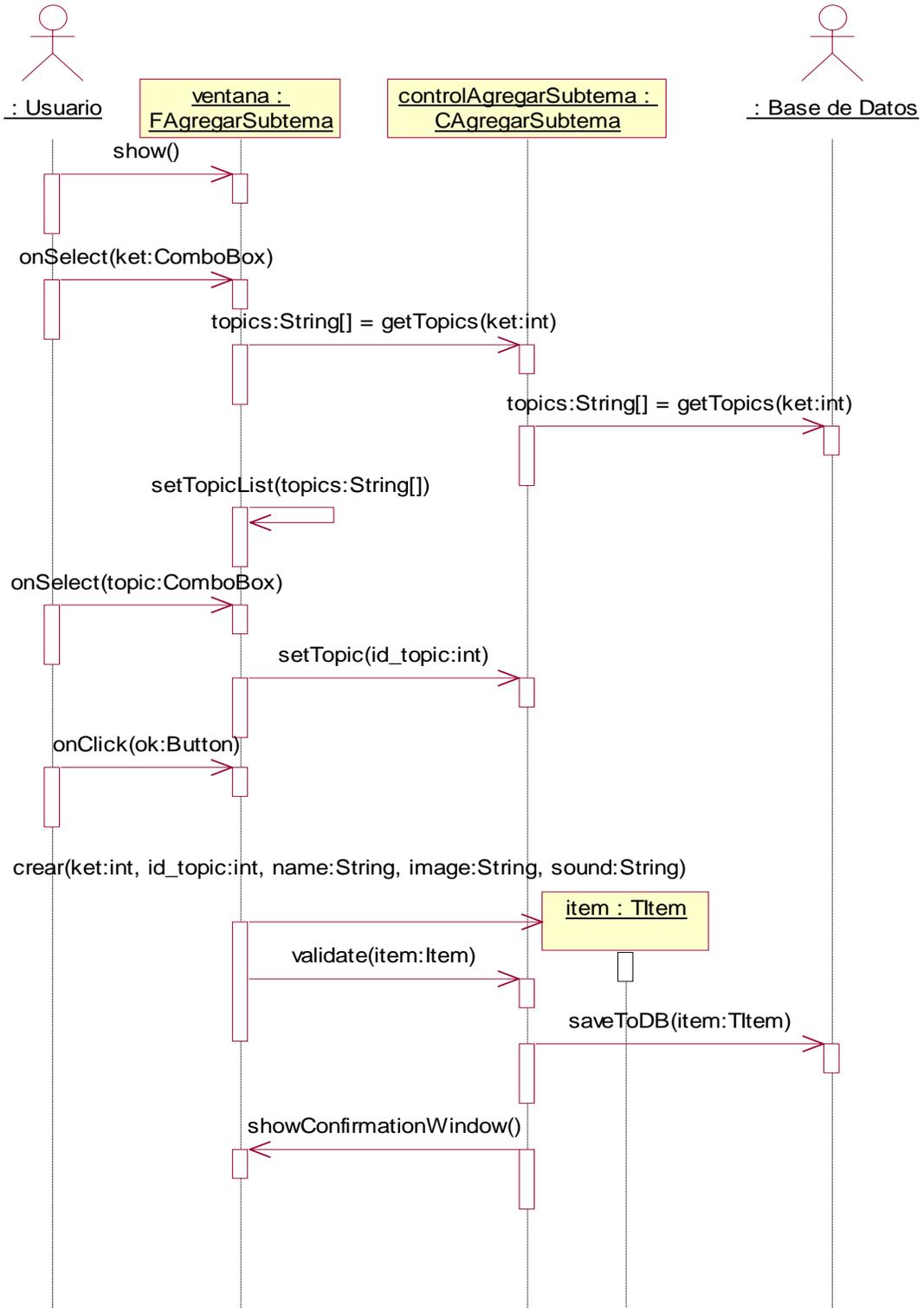
4. Precondiciones

Para que este caso de uso pueda ser realizado por un usuario, éste debe haber agregado un tema.

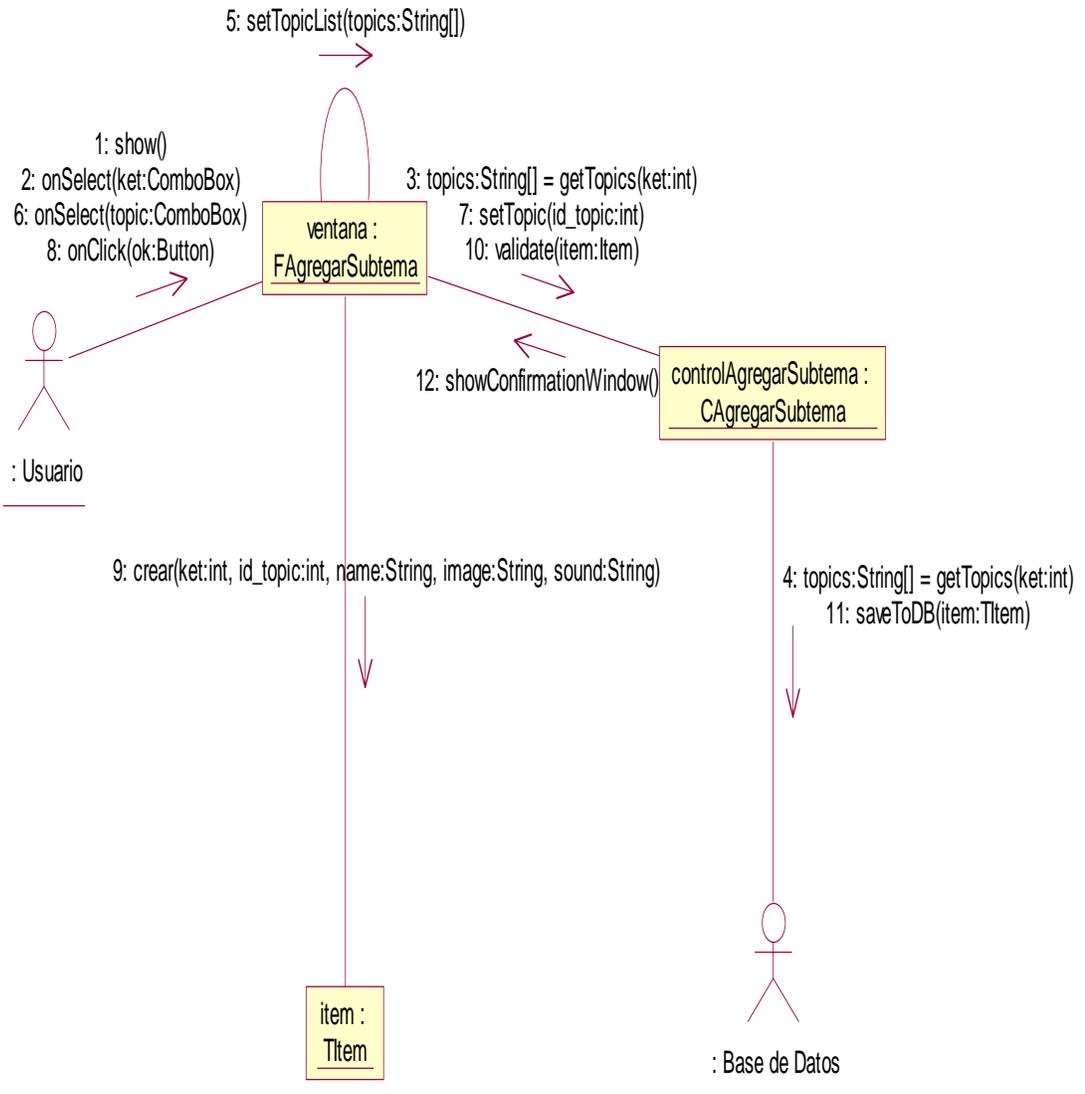
5. Diagrama de clases



6. Diagrama de secuencia del caso de uso *agregar subtema*: Flujo Básico



7. Diagrama de colaboración



3.4.3. Diseño de la base de datos

La base de datos se diseñó con respecto a la información requerida por parte de los profesores. En la Figura 3.17 se muestra el diagrama de la base de datos. Los temas desempeñan un papel muy importante dentro del almacenamiento de datos.

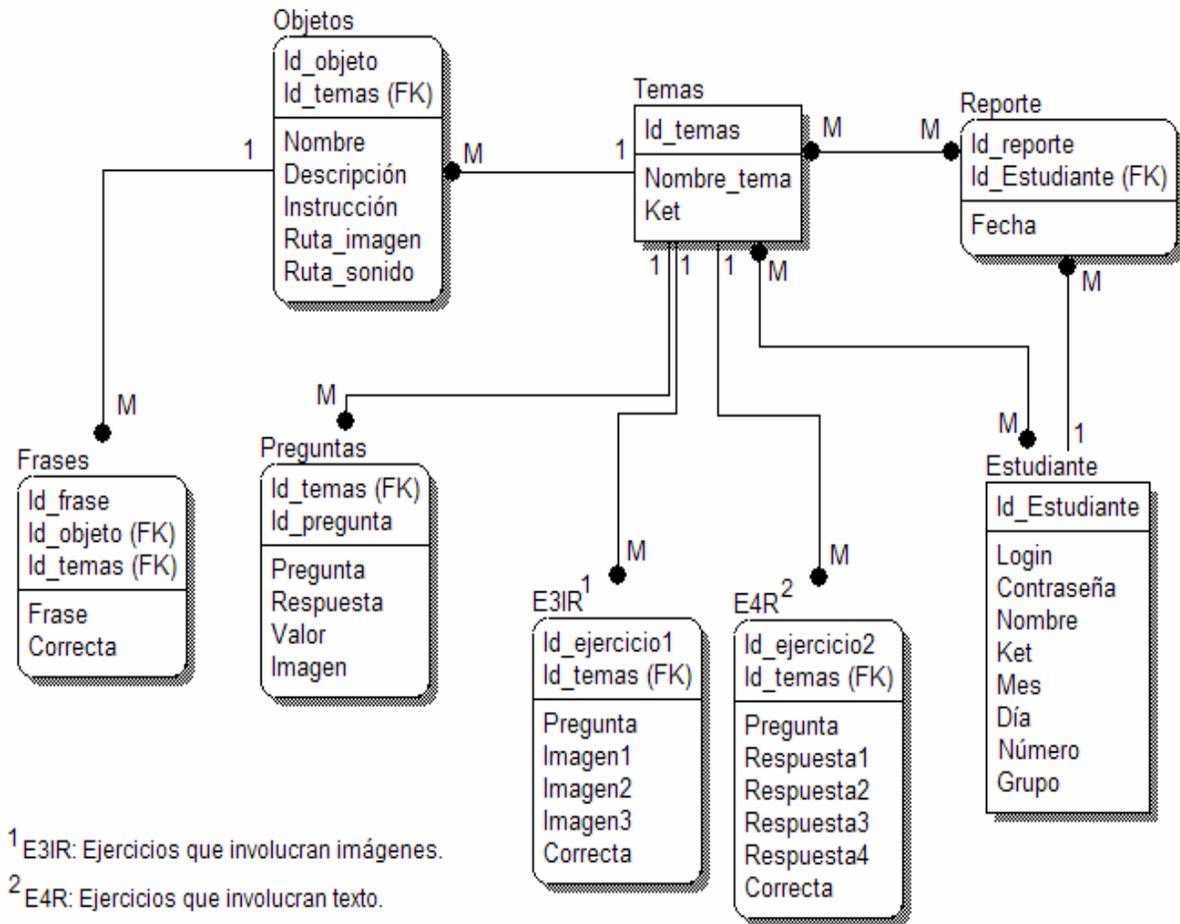


Figura 3.17. Diagrama lógico de la base de datos

Para determinar el diseño de la base de datos se tomaron en cuenta los diferentes tipos de juegos. Cada juego accede a ciertas tablas para obtener la información necesaria. A continuación se describe la forma en que cada tabla está relacionada con cada uno de los juegos:

Ahorcado: Para este juego se utiliza la tabla de temas y objetos.

Cachar palabras: Utiliza la tabla de temas, frases y objetos.

Jugando con los verbos: Este juego utiliza la tabla temas y preguntas.

Relacionar palabras con imágenes: Requiere obtener información de la tabla temas y objetos.

Ejercitador tipo A: Utiliza la tabla de temas y la tabla que involucra imágenes.

Ejercitador tipo B: Este ejercicio utiliza la tabla temas y la tabla que involucra textos.

La función de cada tabla es:

Temas: Almacena los temas y el nivel KET correspondiente a cada tema.

Objetos: La información que se almacena en esta tabla corresponde a los subtemas. Se almacena el nombre del subtema, la descripción de dicho subtema, la instrucción para el juego de atrapar palabras, la ruta en donde se encuentra la imagen asociada a este subtema y la ruta del sonido.

Frases: En esta tabla se almacena la información correspondiente al juego de atrapar palabras. Se almacena si la frase es correcta o incorrecta para poder evaluarla.

Preguntas: Esta tabla contiene la información correspondiente al juego de verbos. Almacena la oración o pregunta correspondiente al verbo, la respuesta a esa pregunta, el valor de esa pregunta u oración y la ruta de la imagen asociada al verbo.

E3IR: La información que se almacena en esta tabla son las preguntas y las rutas donde se encuentran las imágenes, así como la respuesta correcta asociada a cada pregunta.

E4R: La finalidad de esta tabla es guardar la información de las preguntas para el ejercitador tipo B, almacena las posibles respuestas y la respuesta correcta.

Reporte: Almacena información correspondiente a los avances del estudiante y la fecha en que utilizó el sistema.

Estudiante: Contiene la información descriptiva del estudiante como: nombre completo, matrícula, contraseña para iniciar la sesión, nivel de KET, grupo y fecha de nacimiento (mes/día/año), este último dato es de utilidad en caso de que el estudiante olvide su contraseña.

En la fase de implementación que se muestra en el siguiente capítulo, se incorporaron los aspectos del diseño identificados en esta fase. También se presentan las

3. Análisis y diseño del software educativo

imágenes de la versión final del sistema, así como una breve descripción de cada uno de los módulos que lo conforman.

Capítulo 4

4. Implementación de EZ-KET

Antes de implementar y desarrollar el sistema EZ-KET, se realizó un estudio del tipo de herramientas a utilizar con la finalidad de escoger la mejor opción.

4.1 Herramientas de Desarrollo

El lenguaje de programación utilizado para este proyecto fue C++. Este lenguaje soporta la programación orientada a objetos. El ambiente de desarrollo utilizado fue C++ Builder, debido a que proporciona mayor facilidad en el desarrollo de la interfaz comparada con JBuilder. El entorno de desarrollo integrado en C++ Builder permite desarrollo rápido de aplicaciones (RAD).

Por otra parte, el sistema requiere de una base de datos que almacene la información correspondiente a los temas de cada nivel KET, así como la información de cada estudiante. Por tal motivo, se incorporó SQL Server 2000 [URL08] al proyecto; el cual es un paquete completo de análisis y base de datos relacionales. La razón por la que se utilizó este manejador fue porque es altamente reconocido en el ámbito empresarial, el número de casos de éxito⁶ es muy grande. Además SQL Server 2000 es compatible con C++ Builder, permite realizar una conexión fácil entre uno y otro. Otro punto importante es que soporta procedimientos almacenados, lo cual es de gran utilidad para el desarrollo del sistema. Otra razón es que el sistema operativo a utilizar es Windows y SQL Server 2000 está diseñado para trabajar bajo este sistema operativo.

Para el tratamiento de imágenes e interfaz gráfica se utilizó 3D MAX Studio, el cual es un software para animación y modelado en 3D, que permite desarrollar juegos, películas, visualizar proyectos CAD y producir efectos visuales [URL03].

El generador de ayuda utilizado fue Shalomhelp, este software es muy sencillo de utilizar, su funcionamiento es muy similar a un editor de textos. Posee características como: incluir imágenes, enlaces a páginas Web y a direcciones de correo electrónico. Se pueden crear archivos de ayuda en diferentes idiomas, según el idioma del sistema operativo.

4.2 Software Educativo: EZ-KET

En la Figura 4.1 se muestra la forma en que quedó estructurado el software educativo para los estudiantes de nivel KET de la UTM.

⁶ Para verificar los casos de éxito acuda a la página de Microsoft.
<http://www.microsoft.com/latam/sql/evaluation/overview/default.asp>

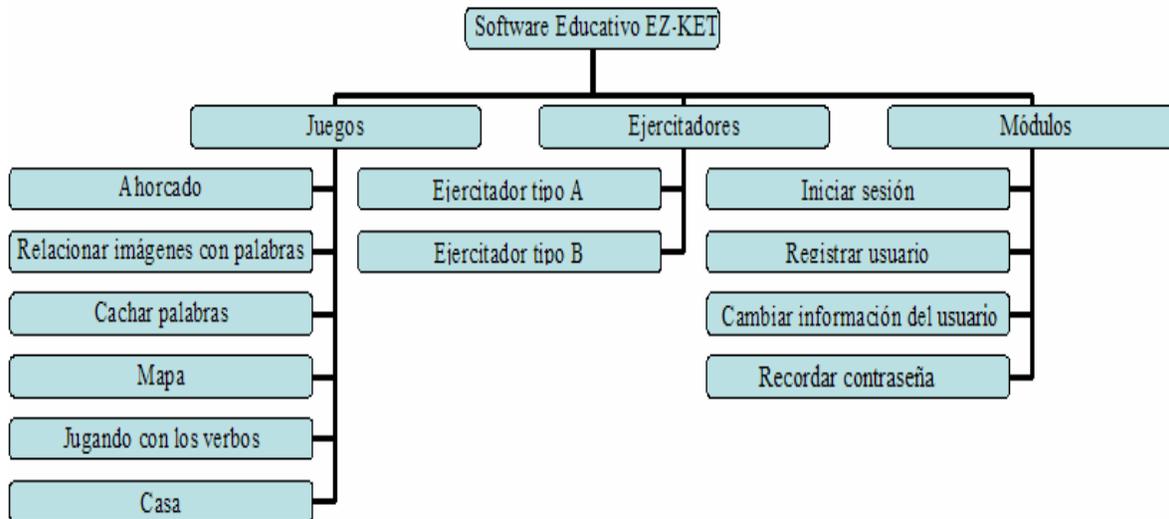


Figura 4.1. Estructura del software educativo EZ-KET.

4.2.1. Módulo: Registro de Usuario

En este módulo el estudiante introduce datos importantes para el control de su sesión en el sistema, los cuales son: matrícula, contraseña (máximo de ocho caracteres), nombre del estudiante, nivel KET, grupo y fecha de nacimiento (mes/día/año), tanto la fecha de nacimiento como la matrícula sirven para que el sistema recupere la contraseña del estudiante en caso de que la haya olvidado. La Figura 4.2 muestra la interfaz de este módulo.

Student Registration

Personal Information

ID Number:

Password:

Re-type Password:

Name:

Last Name:

KET Level:

Group:

If You Forget Your Password

Birthday dd/yyyy

Figura 4.2. Registro del estudiante.

4.2.2. Módulo: Inicio de Sesión

El módulo permite al usuario entrar al sistema. Se debe proporcionar la matrícula del estudiante y su contraseña. La Figura 4.3 muestra la interfaz correspondiente a este módulo.

Sign In

ID Number:

Password:

[Forgotten your password?](#)

[Change information](#)

If you are new, [Sign Up Now](#)

Figura 4.3. Inicio de sesión.

Una vez registrado el usuario, se muestra la pantalla de inicio del sistema EZ-KET. Esta pantalla refleja un juego de serpientes y escaleras como se muestra en la Figura 4.4.

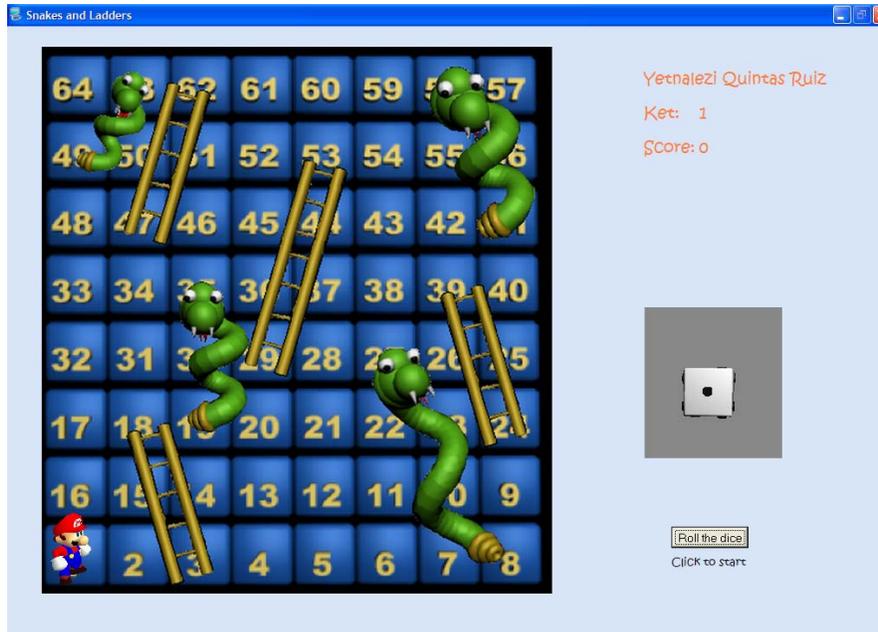


Figura 4.4. Juego Serpientes y Escaleras.

Antes de que el sistema fuera empleado por los usuarios finales, se sometió a nuevas pruebas de usabilidad, éstas fueron realizadas por un usuario experto, quién interactuó con el sistema y al final dio sus observaciones. Las observaciones realizadas a la pantalla anterior fueron:

- Mostrar el dado que aparece al inicio del juego en tercera dimensión para dar mayor realce al mismo.
- Poder configurar el volumen desde el inicio de la aplicación.
- Agregar un botón de salir.
- Cambiar el color de las letras del nombre del estudiante, nivel KET y puntuación.
- Cambiar el color de fondo de acuerdo a los niveles KET, entre otras.

Las observaciones fueron tomadas en cuenta quedando implementadas en la versión final de EZ-KET. En las Figuras 4.5, 4.6 y 4.7 se muestran los cambios resultantes.

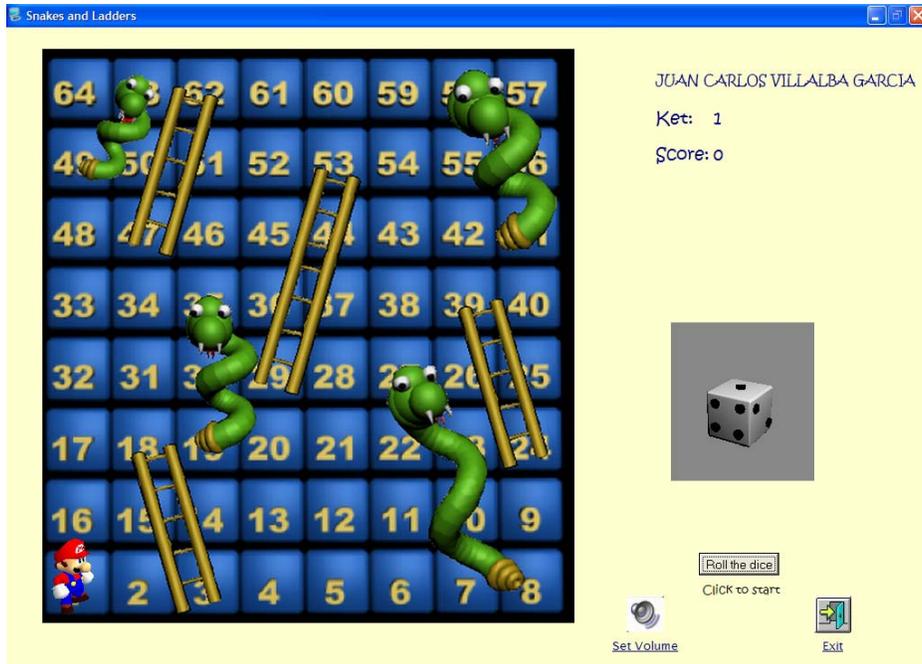


Figura 4.5. Juego Serpientes y Escaleras para KET 1 versión final.

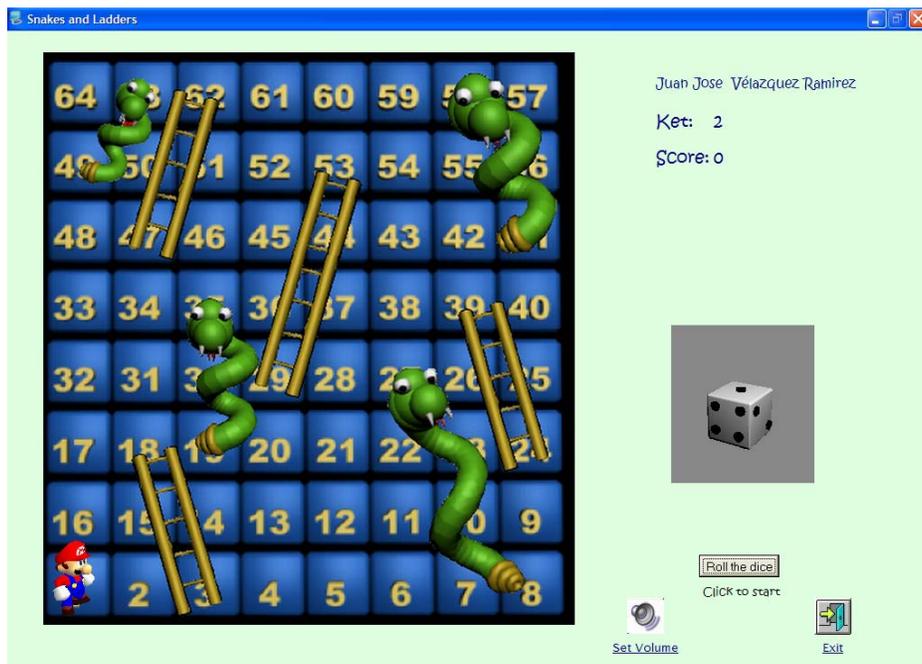


Figura 4.6. Juego Serpientes y Escaleras para KET 2 versión final.

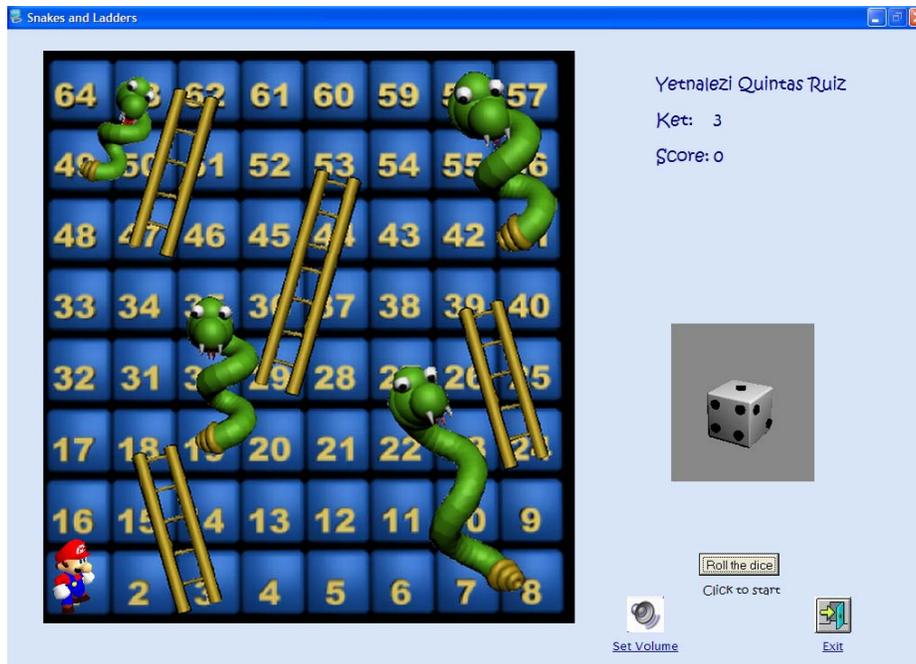


Figura 4.7. Juego Serpientes y Escaleras para KET 3 versión final.

4.2.3. Módulo: Recordar Contraseña

Este módulo se encarga de proporcionar al estudiante su contraseña en caso de que la haya olvidado. El estudiante debe introducir su matrícula y fecha de nacimiento para que el sistema le proporcione su contraseña (ver Figura 4.8).

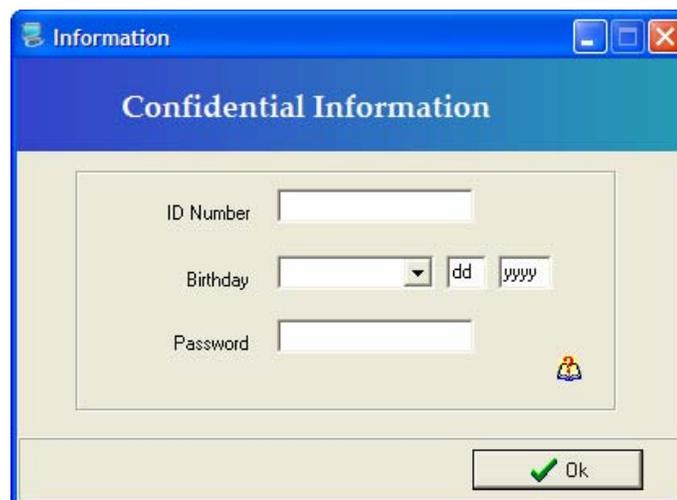


Figura 4.8. Recordar contraseña.

4.2.4. Módulo: Cambiar Información del Estudiante

En este módulo el estudiante puede cambiar su nivel KET y grupo, ya sea porque quiera practicar otros niveles o porque según las evaluaciones realizadas por el Centro de Idiomas de la UTM, el estudiante haya acreditado el pase a otro nivel KET. El cambio de grupo se consideró ya que cuando cambie de semestre el grupo es diferente y se tiene que actualizar dicha información. La Figura 4.9 muestra la imagen correspondiente a este módulo tal y como aparece en el sistema.

Figura 4.9. Interfaz para cambia el nivel KET y/o grupo.

4.2.5. Módulo: Ayuda

La ayuda para el sistema EZ-KET se implementó tanto en Español como en Inglés, ya que las pruebas de usabilidad mostraron que la mayoría de los estudiantes requerían la ayuda en español, pero como se trata de un software que ayuda a practicar el idioma inglés, es necesario presentar una alternativa que incorpore la práctica de este idioma. Como la ayuda se presenta en ambos idiomas, es elección de cada estudiante el idioma en que la consultará. En la Figura 4.10 se muestra un ejemplo de dicho módulo.

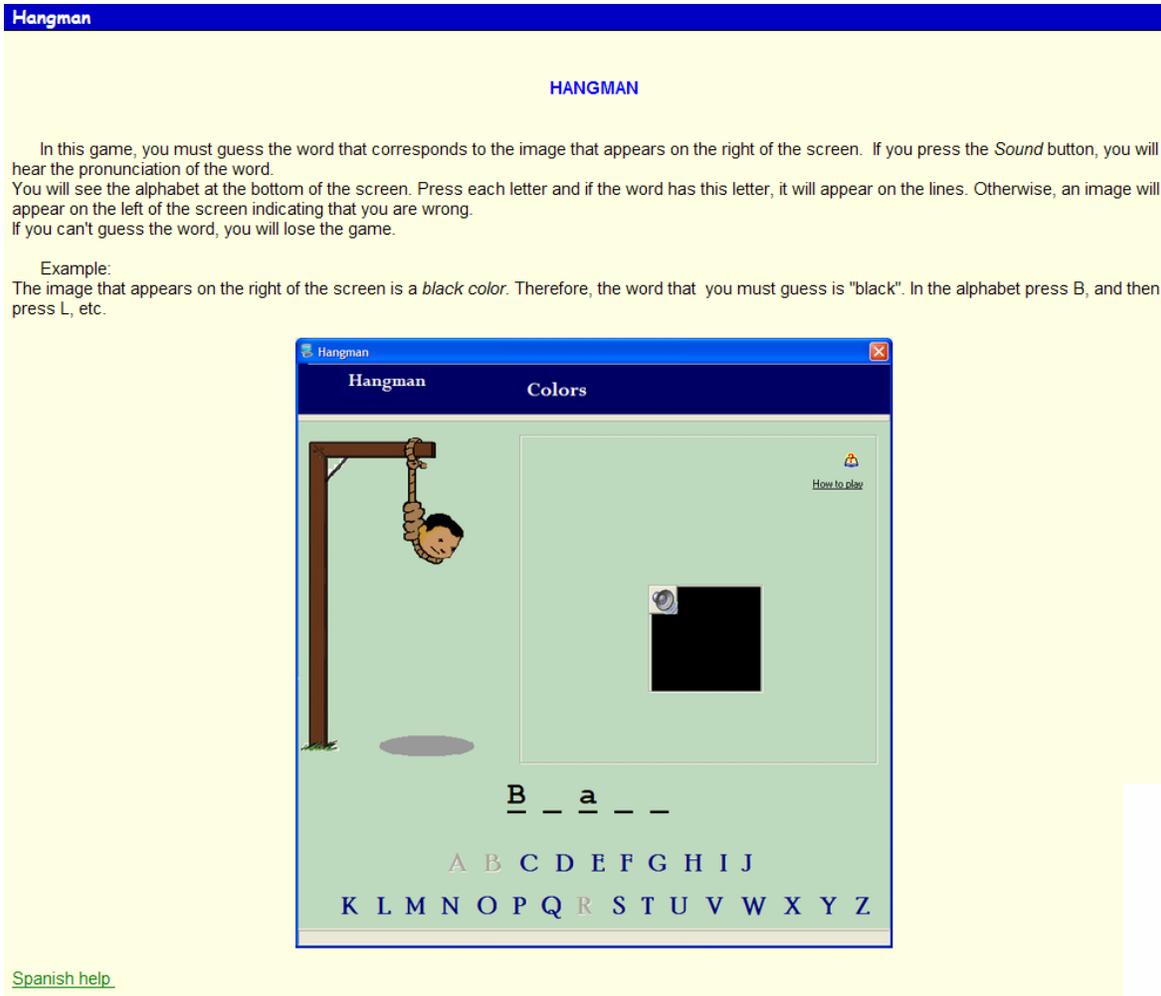


Figura 4.10. Ayuda implementada en el sistema EZ-KET.

4.2.6. Juego: Ahorcado

Este juego permite que el estudiante practique su vocabulario, así como, la forma correcta en la cual se escribe dicha palabra. El estudiante puede también practicar su comprensión auditiva. Para que esto funcione, se debe presionar el botón que tiene una bocina y el sonido de la palabra se escuchará. Al final del juego se escucha el sonido de la palabra. En la parte inferior de la ventana se muestra el abecedario y para que las letras aparezcan en las líneas se deben presionar dichas letras. Si la letra que se presiona se encuentra dentro de la palabra, ésta aparecerá en la línea. En caso contrario aparecerá la imagen de una parte del cuerpo humano del lado izquierdo.

Para KET 1 (Figura 4.11) se tiene una imagen con respecto al tema a practicar del lado derecho de la pantalla.



Figura 4.11. Juego del ahorcado para KET 1.

Para KET 2 aparece la descripción de la palabra a adivinar del lado derecho de la pantalla, como se muestra en la Figura 4.12.



Figura 4.12. Juego del ahorcado para KET 2.

Para KET 3 sólo se cuenta con el audio de la palabra a adivinar. La interfaz aparece en la imagen 4.13.

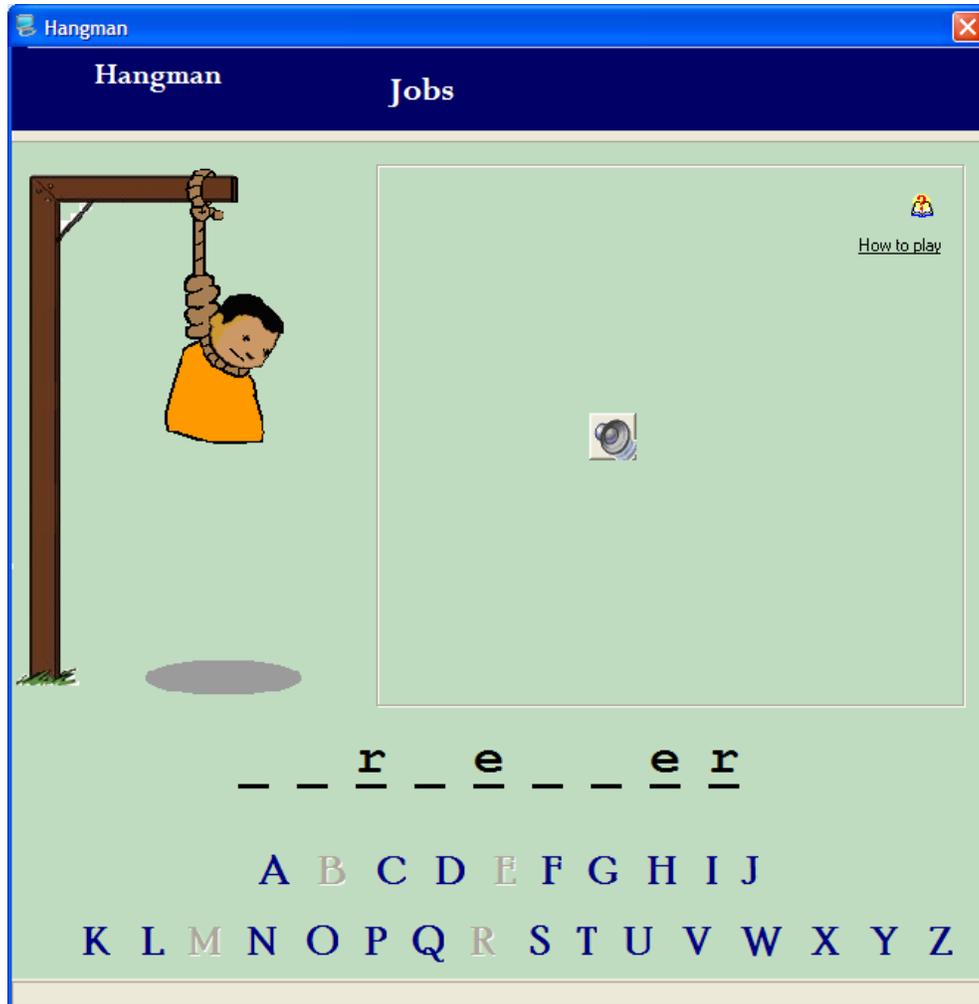


Figura 4.13. Juego del ahorcado para KET 3.

4.2.7. Juego: Relacionar Imágenes con Palabras

Diseñado para que los estudiantes practiquen vocabulario y comprensión auditiva, ya que, cada imagen que se presenta cuenta con su respectivo sonido. Se debe presionar el botón que tiene una bocina para que el sonido se escuche.

El juego consiste en arrastrar las palabras que se presentan en el lado derecho de la ventana hacia las imágenes que se muestran en el lado izquierdo. Al terminar este juego, las respuestas correctas se muestran resaltadas con color verde, mientras que las respuestas incorrectas se resaltan con color rojo. No se especifica cual es la palabra correcta cuando el estudiante se equivoca, los profesores sugirieron que no se mostrara la respuesta correcta para motivar a que los estudiantes la investiguen. En la Figura 4.14 aparece la interfaz del juego.

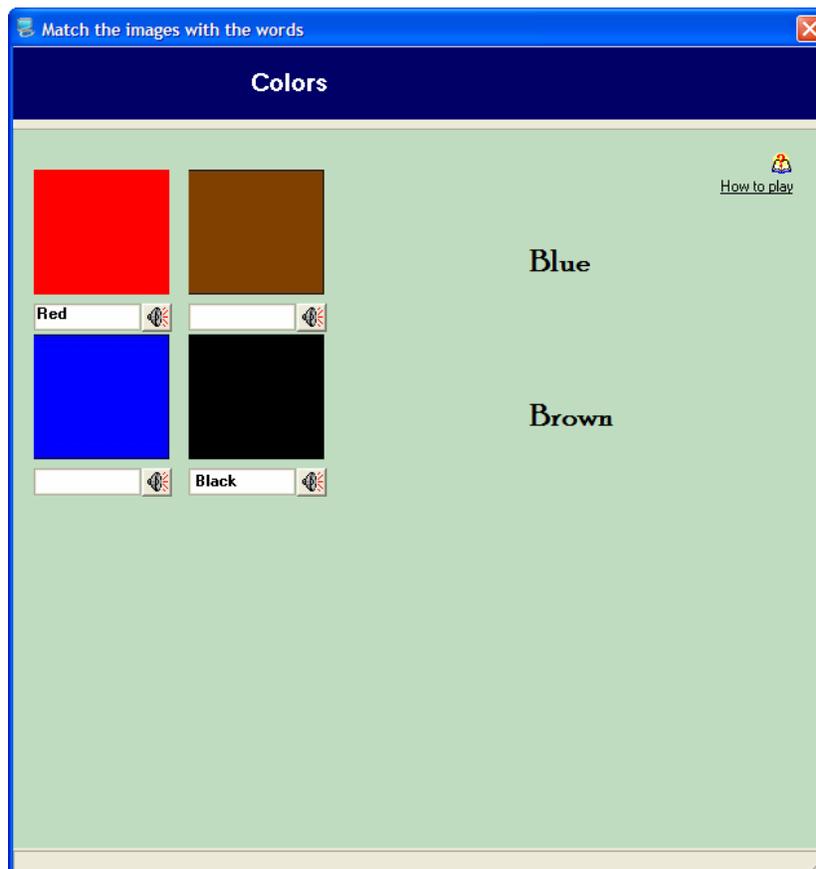


Figura 4.14. Juego de relacionar imágenes con palabras.

4.2.8. Juego: Cachar Palabras

Se considera el tiempo de caída de las palabras o frases. Para cachar las palabras se debe mover el carrito de supermercado, si se atrapa una palabra que no corresponde a la imagen que aparece en el lado izquierdo, ésta aparecerá en el cuadro inferior que se encuentra en el lado derecho, en el caso contrario aparecerá en el lado izquierdo, indicando que la respuesta es correcta. La velocidad con la que caen las palabras varía según el nivel KET. La Figura 4.15 muestra este juego.

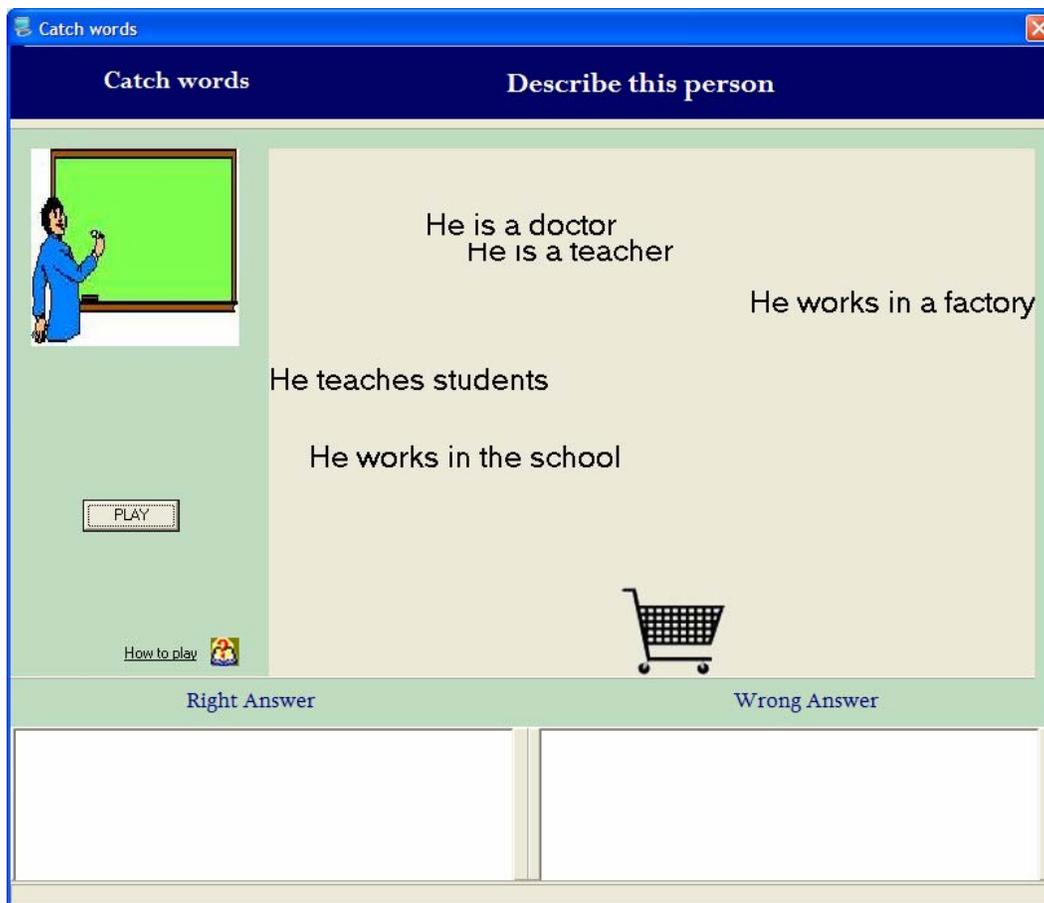


Figura 4.15. Juego de cachar palabras.

4.2.9. Juego: Mapa

El estudiante debe seguir instrucciones, las cuales aparecen en la pantalla (sólo KET 1 y KET 2); esta interfaz aparece en la Figura 4.16. Para KET 3 sólo se cuenta con el audio, para que el estudiante escuche la instrucción. De acuerdo a una lista de opciones el estudiante debe seleccionar a qué lugar llegó después de seguir las instrucciones presentadas por el juego. La interfaz se muestra en la Figura 4.17.

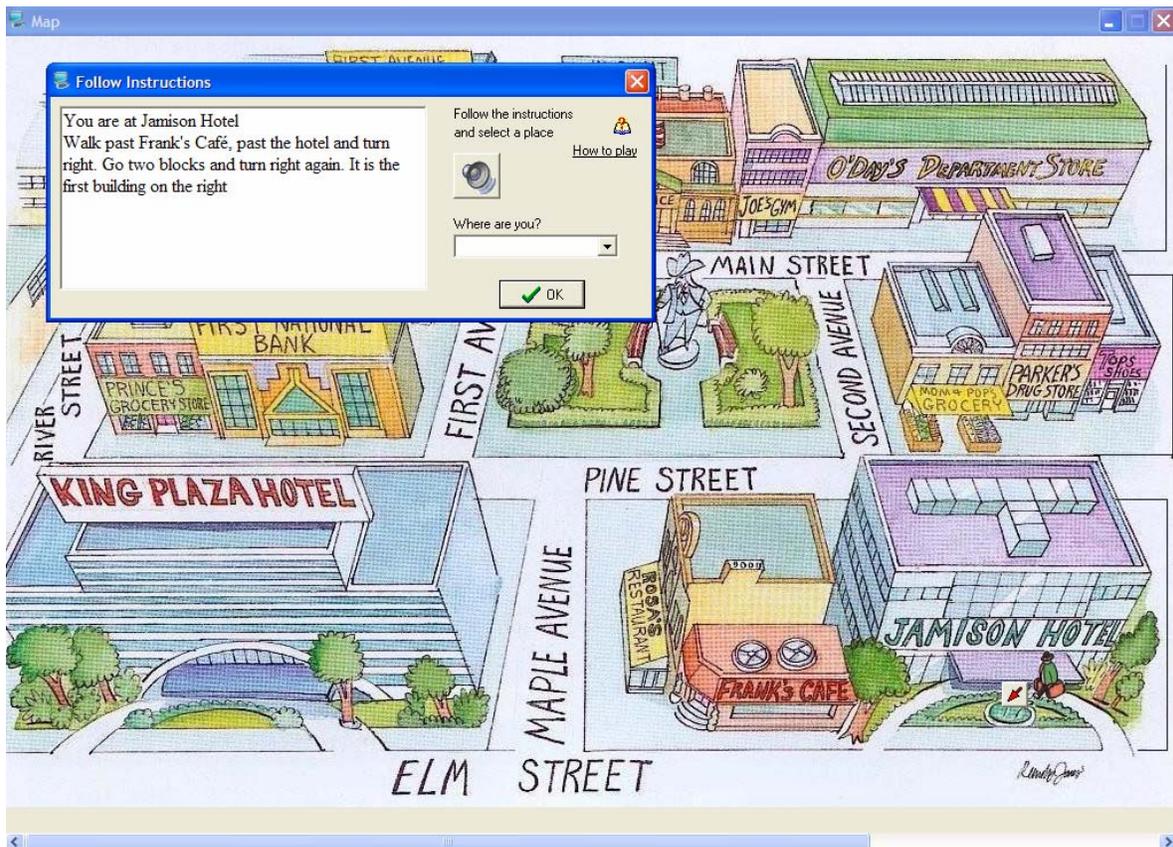


Figura 4.16. Juego del mapa para KET 1 y KET 2.

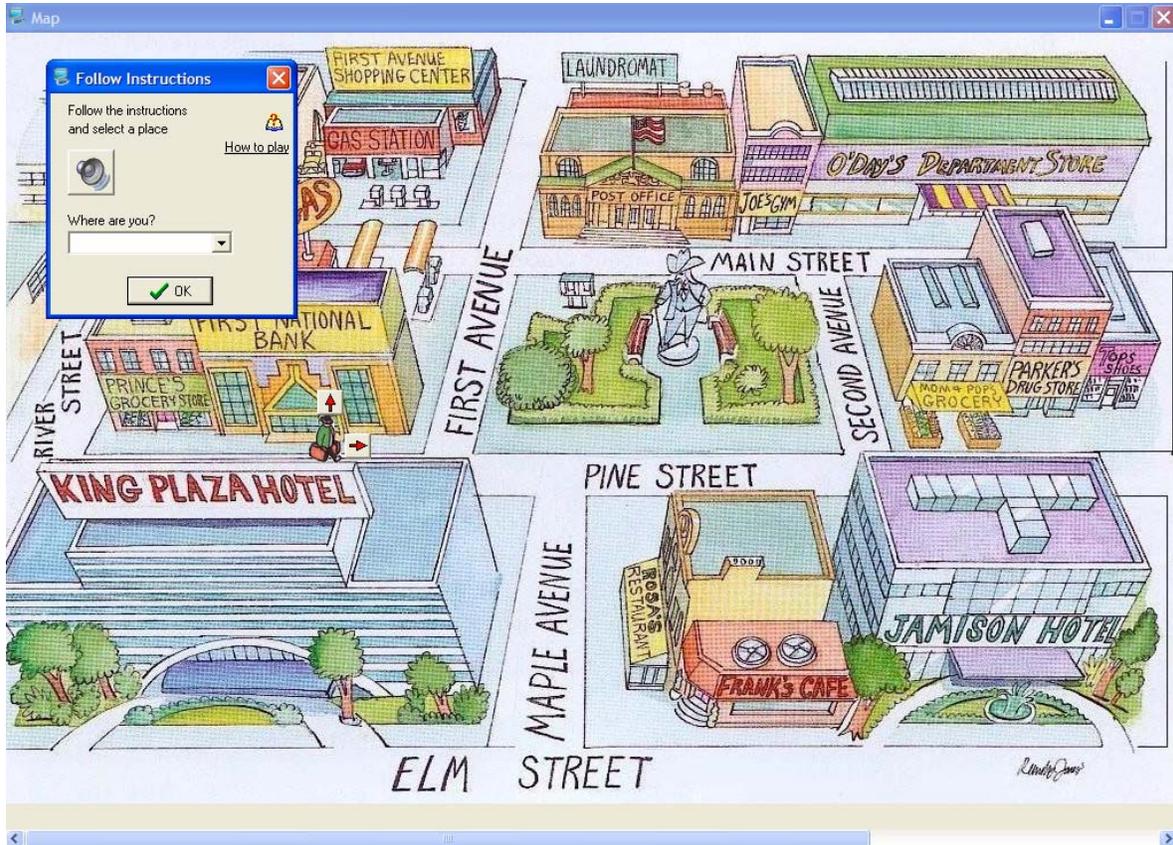


Figura 4.17. Juego del mapa para KET 3.

4.2.10. Juego: Jugando con los Verbos

Como se citó en el apartado anterior, este juego, sólo es para dos niveles de KET (dos y tres), debido a que en el temario no se contempla verbos en pasado para el primer nivel KET. Se utiliza principalmente para practicar verbos en pasado, aunque puede ser usado para otros tiempos gramaticales.

Este juego consiste en presionar cualquier botón de color verde. Los botones tienen un número asociado, dicho número sirve para determinar la complejidad del verbo. En la parte de arriba se especifica la letra con la que empieza el verbo. En la Figura 4.18 se muestra esta ventana. Una vez presionado el botón aparece la ventana de la Figura 4.19, la cual contiene una pregunta. Se debe de escribir la respuesta en esta ventana.



Figura 4.18. Jugando con verbos para KET 2 y KET 3.

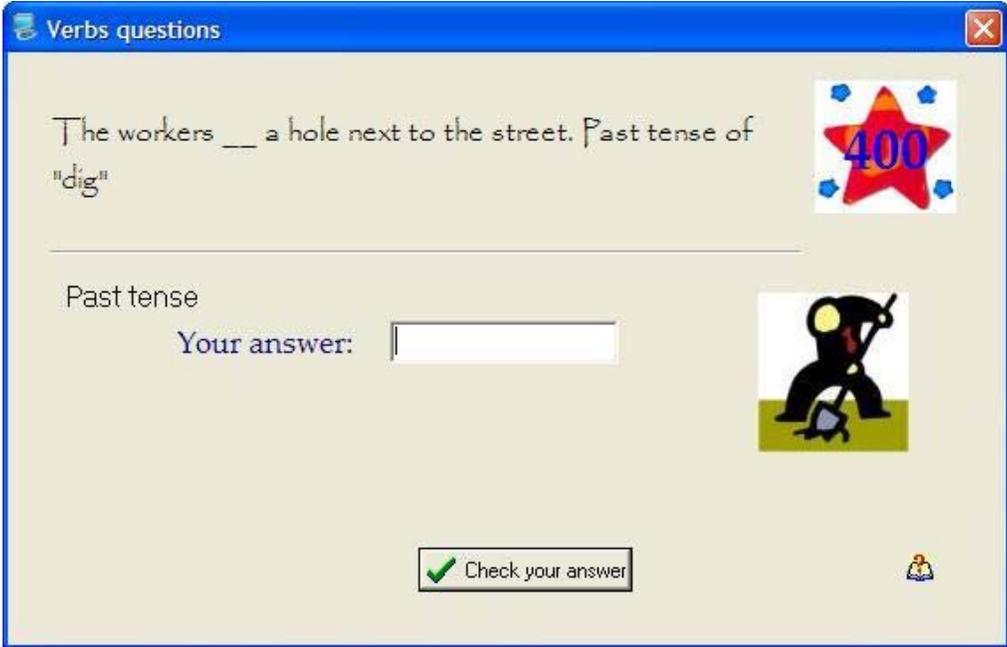


Figura 4.19. Jugando con verbos para KET 2 y KET 3.

4.2.11. Juego: Describe, localiza partes y objetos de una casa

Uno a uno se muestran los cuartos que forman una casa. Para KET 1 se debe seleccionar el nombre de cada cuarto. En KET 2 se implementó una interfaz en donde el estudiante pudiera arrastrar las palabras que se encuentran del lado derecho de la ventana hacia la imagen correspondiente. La Figura 4.20 muestra la interfaz de este juego. Finalmente, para KET 3 se implementó un juego en donde el estudiante escribe los nombres de los muebles correspondientes a cada habitación.

Todas las modalidades de este juego contienen sonido, de forma que el estudiante practique su comprensión auditiva. Para escuchar el sonido se debe presionar el botón que muestra una bocina.

En este juego, el usuario experto hizo las siguientes observaciones:

Cambiar el color de las flechas de rojo a verde, dado que el color rojo indica “alto” y el usuario podría confundirse o tener una mala idea del uso del botón *Next*

Subir el botón *Next*

- Agregar la instrucción en la parte central y arriba de la imagen

La ventana final para el nivel KET 2 se muestran en la Figura 4.21.

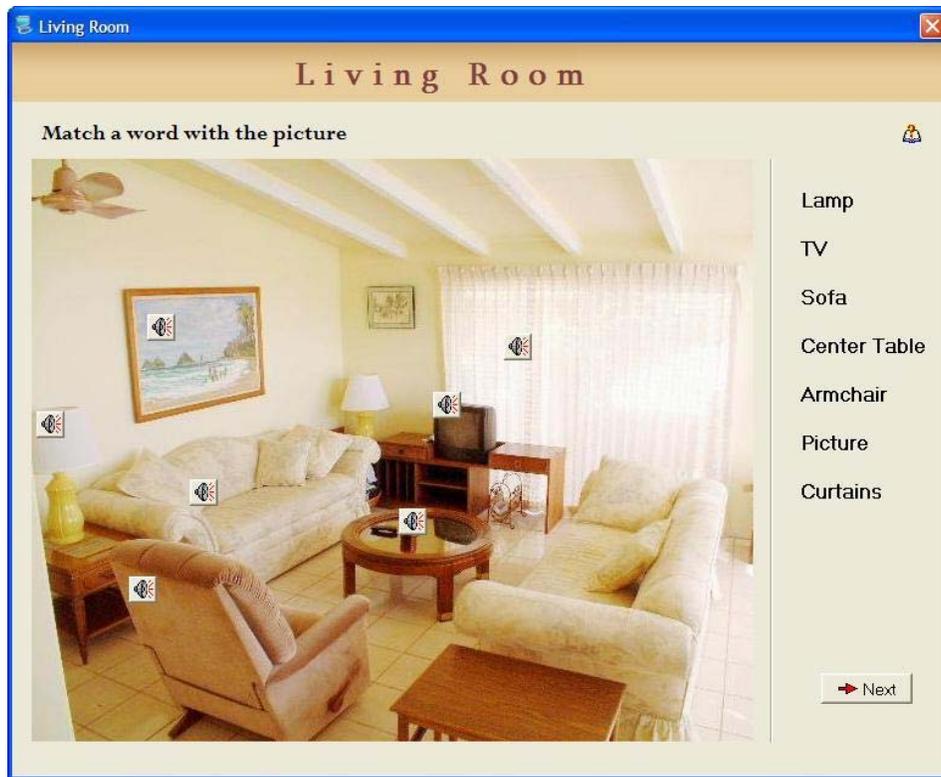


Figura 4.20. Juego de la casa para KET 2.

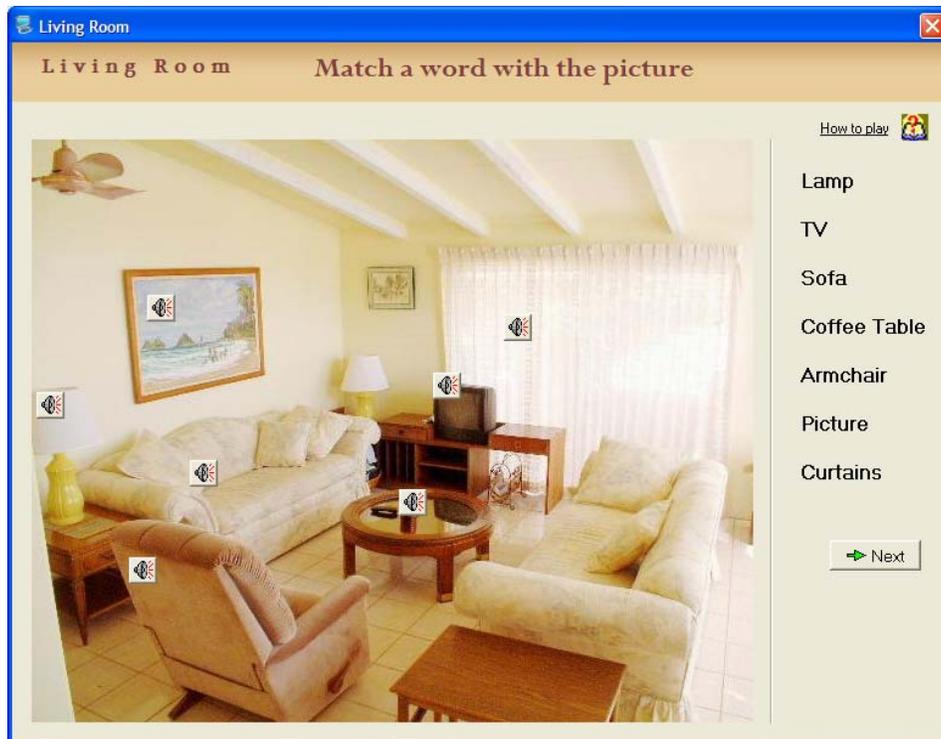


Figura 4.21. Juego de la casa para KET 2 versión final.

4.2.12. Ejercitadores tipo A y tipo B

Basados en las ideas de los profesores, sirven para evaluar en la forma tradicional el conocimiento de los estudiantes. Estos ejercicios tienen un estilo similar a los que se presentan en los exámenes. Aparece una pregunta en la parte de arriba y se tiene que seleccionar una opción. Las interfaces de los ejercicios de este estilo se muestran en las Figuras 4.22 y 4.23.

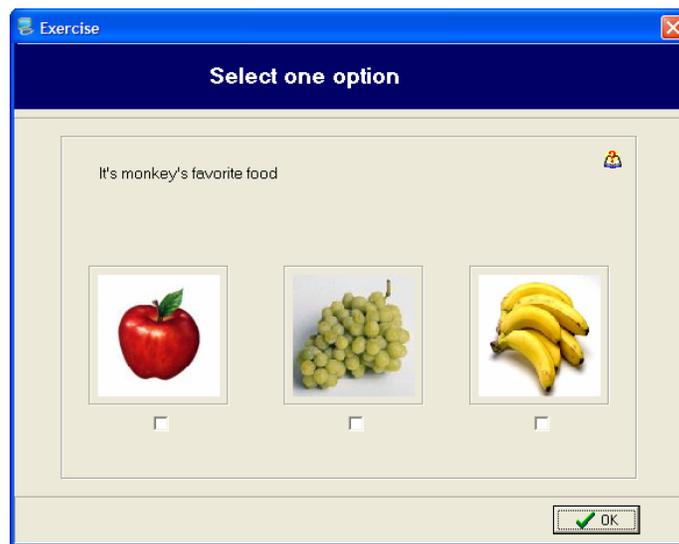


Figura 4.22. Ejercitador tipo A.

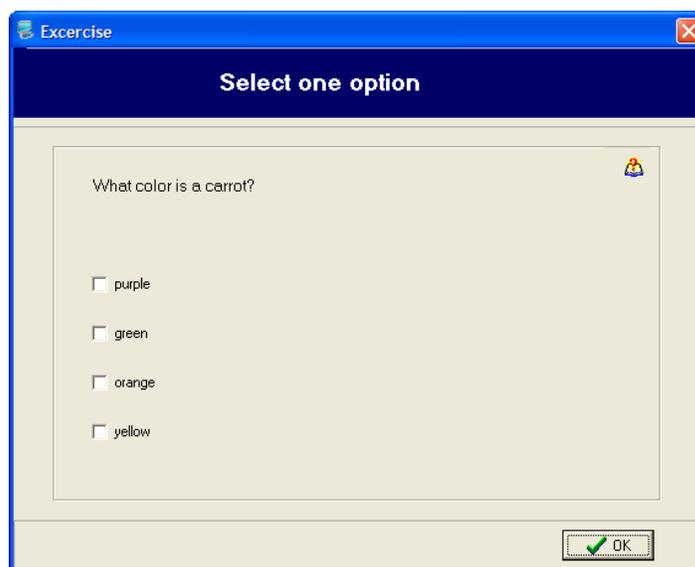


Figura 4.23. Ejercitador tipo B.

Finalmente, el usuario experto hizo algunas observaciones más como son:

1) Agregar un splash screen ⁷

Se agregó una pantalla de bienvenida al sistema Figura 4.24, esta pantalla se despliega al inicio de EZ-KET por cinco segundos.



Figura 4.24. Splash screen.

2) Cambiar la presentación de la ventana de aviso

El usuario experto sugirió cambiar los mensajes de las ventanas de aviso, permitiendo girar el dado desde la ventana que manda el mensaje cuando no se tiene que realizar un juego o ejercicio. Estas observaciones se llevaron a cabo para que el usuario entendiera mejor los avisos y evitara presionar un botón de más. En la Figura 4.25 aparece la ventana previa a los cambios y la Figura 4.26 muestra la versión final de dicha ventana.

⁷ Término computacional para referirse a una imagen que aparece mientras un programa se está cargando en memoria.



Figura 4.25. Ventana previa a la versión final.



Figura 4.26. Ventana de aviso final.

3) Agregar sonido cuando “Mario Bros.” sube la escalera y cuando cae de la serpiente

Estas observaciones no se implementaron debido a que el equipo de cómputo en el que estará residente el sistema EZ-KET se ejecutaría demasiado lento provocando desesperación en los usuarios.

4.3 Sistema de Retroalimentación

Este sistema sirve para introducir nuevos datos correspondientes a los niveles KET, permite modificar dicha información y consultar los reportes de los estudiantes. Este software consta de 11 módulos. Es importante aclarar que este sistema es usado solamente por usuarios de habla inglesa, por lo tanto, los módulos se muestran en idioma Inglés. La interfaz principal se muestra en la Figura 4.27.

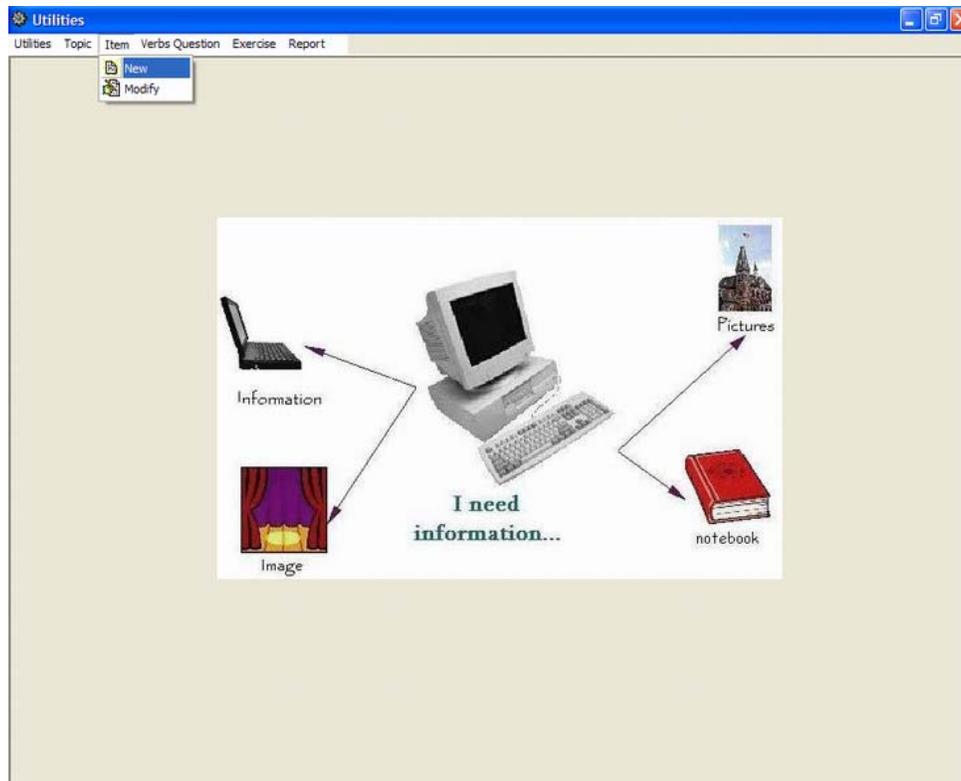


Figura 4.27. Interfaz principal del sistema de retroalimentación.

4.3.1. Módulo: New Topic

Este módulo sirve para que los profesores agreguen un nuevo tema. Se debe de seleccionar un nivel KET y luego escribir el nombre del tema. Este módulo no sufrió cambios, ya que la forma en que se presentan los datos les pareció fácil y entendible a los profesores. La interfaz implementada se muestra en la Figura 4.28.

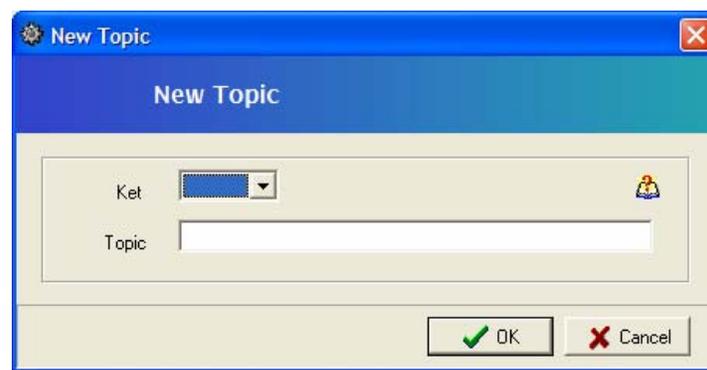


Figura 4.28. Pantalla para agregar un tema.

4.3.2. Módulo: Modify a Topic

Si el profesor se equivoca al escribir el nombre del tema o desea cambiarlo por cualquier otro motivo, puede hacerlo con ayuda de este módulo. Se debe proporcionar el nivel KET, después se selecciona el tema que se desea modificar y finalmente se escribe el nuevo nombre del tema.

Este módulo no sufrió cambios, porque de igual forma a los profesores les pareció adecuado. En la Figura 4.29 se puede observar la interfaz final.

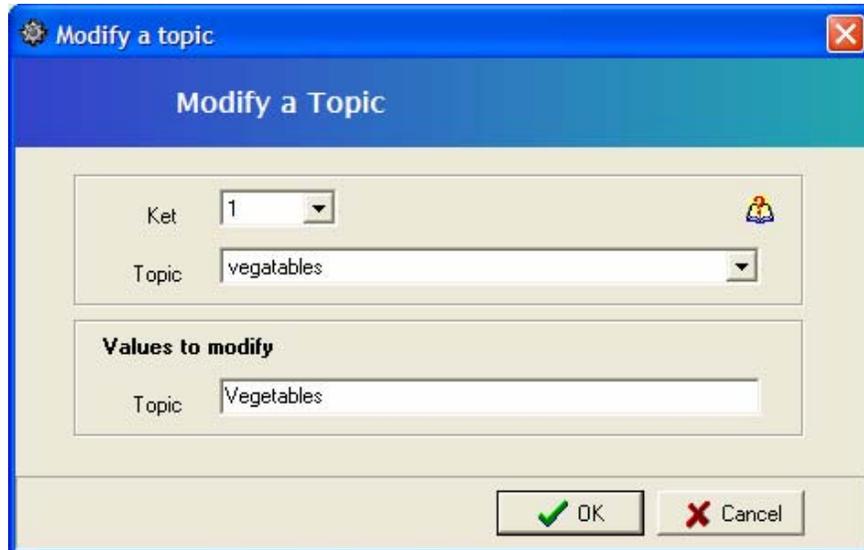


Figura 4.29. Pantalla para modificar un tema.

4.3.3. Módulo: New Item

Este módulo ayuda a agregar un elemento asociado a un tema. Se debe seleccionar un nivel KET, luego un tema y después escribir toda la información relacionada con el elemento como es: nombre, descripción (sólo se proporciona para KET 2), instrucción (se utiliza en el juego de atrapar palabras, con esto el estudiante sabe qué debe hacer con las palabras que van cayendo), se debe seleccionar una imagen asociada a este elemento, además de un archivo de audio. Si se presiona el botón con la leyenda “*Test*” el profesor puede escuchar el audio del archivo que seleccionó. También se deben proporcionar palabras o frases asociadas a este elemento y seleccionar si éstas son correctas o incorrectas, esta información es utilizada por el juego de atrapar palabras.

En este módulo los profesores sugirieron cambios en las etiquetas. Ejemplo, la etiqueta tenía el nombre de “*sentence*” y los profesores pidieron que se cambiara a “*utterance*”. La interfaz asociada a este modulo a parece en la Figura 4.30.

Figura 4.30. Pantalla para agregar un elemento asociado a un tema.

4.3.4. Módulo: Modify an Item

Si se necesita cambiar información correspondiente a un elemento asociado a un tema agregado, éste es el módulo indicado. Se debe seleccionar un nivel KET, un tema y el elemento que se desee modificar.

El cambio que se presentó en este módulo fue el de la etiqueta “*sentence*” por la etiqueta “*utterance*”. La interfaz implementada se muestra en la Figura 4.31.

Figura 4.31. Pantalla para modificar un elemento asociado a un tema.

4.3.5. Módulo: Create a New Question for the Verbs Game

Este módulo fue diseñado para incluir temas referentes a verbos. Se debe agregar un tema para ser utilizado. Primero se selecciona el nivel KET (sólo KET 2 y KET 3), después se selecciona un tema, a continuación se escribe una oración o pregunta correspondiente a un tema de verbos, se debe seleccionar una imagen asociada a la pregunta u oración escrita, se debe de escribir la respuesta (verbo en pasado) y por último seleccionar un puntaje para la pregunta dependiendo del grado de complejidad.

Este módulo no sufrió cambios, ya que les resultó fácil la forma de capturar la nueva información. La Figura 4.32 muestra la interfaz.

The screenshot shows a software window titled "Create a new question for the Verbs Game". The window has a blue header bar with the title. Below the header, the main content area is light beige. At the top of this area, the text "Create a new question for the verbs game" is displayed. The interface contains several input fields and buttons:

- Ket:** A dropdown menu with the value "3" selected.
- Topic:** A dropdown menu with the value "Verbs starting with C" selected.
- Utterance:** A text input field containing "My mom __ a cake. Past tense of 'Cook'".
- Image 1:** A text input field containing "C:\Images\Verbs\cook.jpg" and a "Search" button to its right.
- Answer:** A text input field containing "Cooked".
- Value:** A dropdown menu with the value "200" selected.
- Image Preview:** A small window showing a cartoon chef holding a fork, with a "Clear" button below it.
- Buttons:** "OK" and "Cancel" buttons are located at the bottom right of the window.

Figura 4.32. Pantalla para agregar una pregunta u oración de verbos.

4.3.6. Módulo: Modify a Question for the Verbs Game

La finalidad de este módulo es modificar la información correspondiente a las preguntas de los verbos. Se debe seleccionar un nivel KET, un tema y la pregunta u oración a modificar. La interfaz final se muestra en la Figura 4.33.

Figura 4.33. Pantalla para modificar una pregunta u oración de verbos.

4.3.7. Módulos: Create a New Exercise type A and type B

La información que se necesita para agregar un ejercicio tipo A es el nivel KET, el tema, escribir la pregunta, tres imágenes y seleccionar cuál es la imagen correcta relacionada con la pregunta; en la Figura 4.34 se presenta la interfaz correspondiente. En el módulo para agregar un ejercicio tipo B se necesita seleccionar el nivel KET, el tema, escribir la pregunta y también escribir cuatro posibles respuestas y al final seleccionar cuál de ellas es la correcta; la interfaz implementada se muestra en la Figura 4.35.

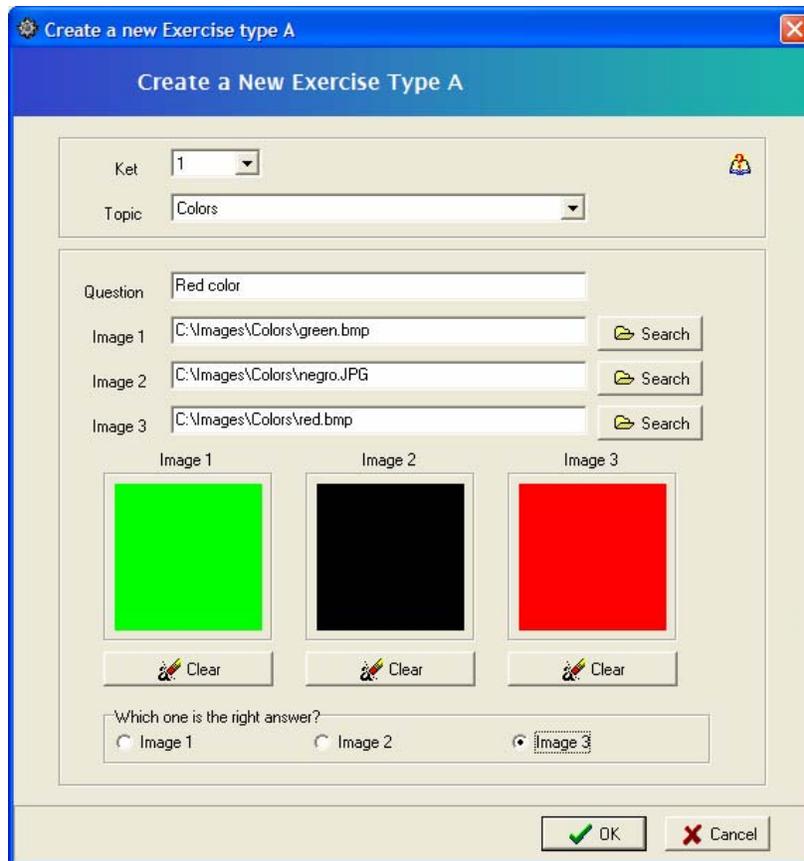


Figura 4.34. Pantalla para agregar un ejercitador tipo A.

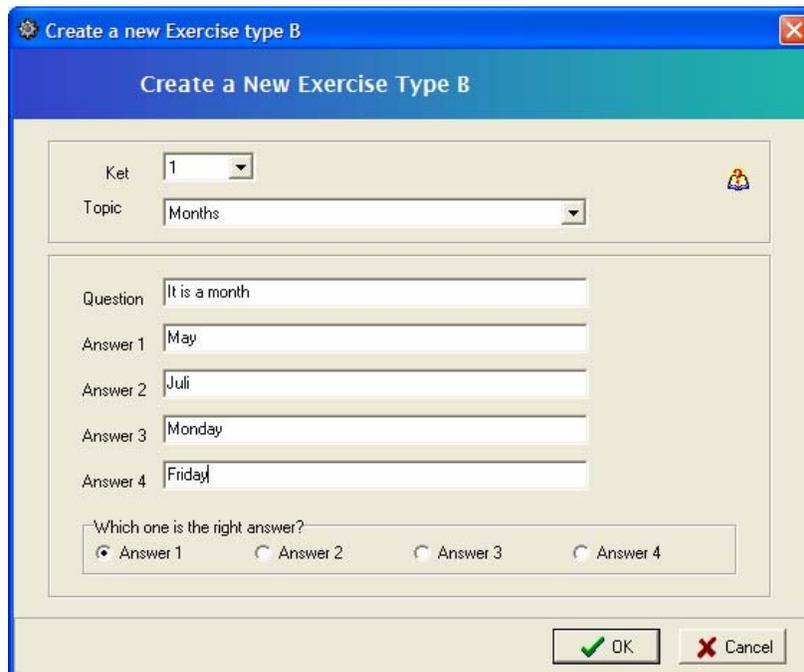


Figura 4.35. Pantalla para agregar un ejercitador tipo B.

4.3.8. Módulos: Modify an exercise type A and modify an exercise type B

Si los profesores necesitan cambiar la información relacionada con los ejercicios tipo A o tipo B, se debe seleccionar el nivel KET, el tema y la pregunta. La información aparece en la parte inferior de la ventana donde los profesores pueden realizar los cambios. Las interfaces se muestran en las Figuras 4.36 y 4.37.

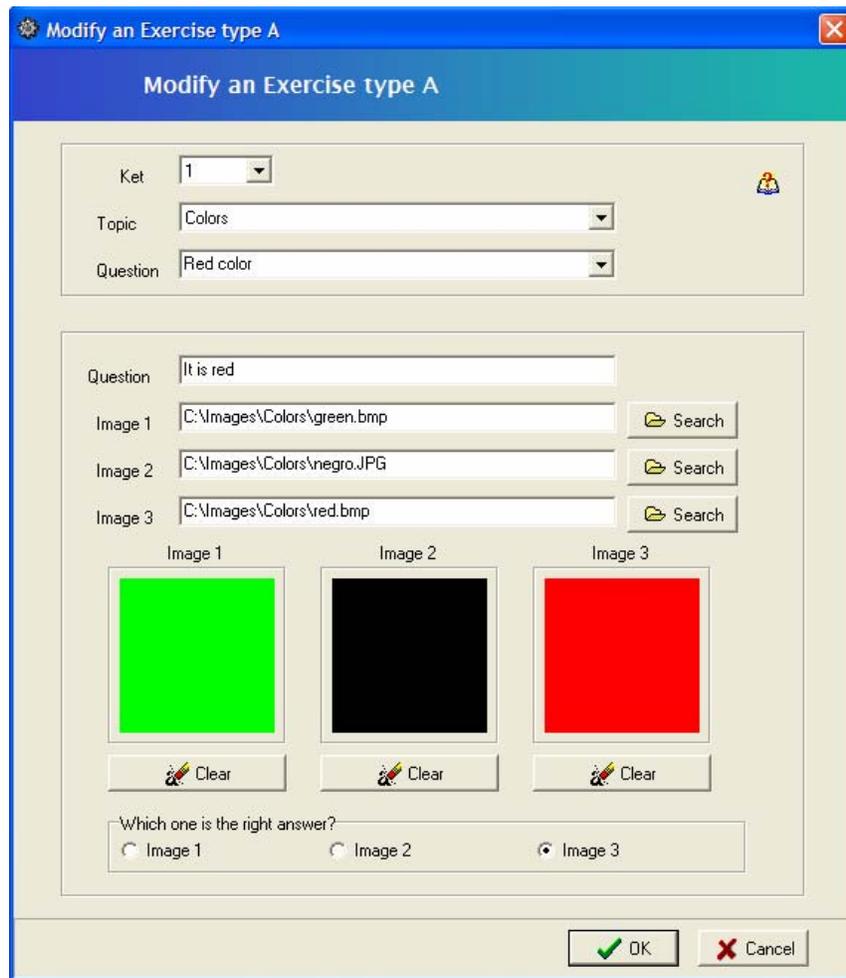


Figura 4.36. Pantalla para modificar un ejercitador tipo A.

Figura 4.37. Pantalla para modificar un ejercitador tipo B.

4.3.9. Módulo: Report

Este módulo sirve para que los profesores revisen el desempeño de los estudiantes. Para buscar un reporte, el profesor debe seleccionar un nivel KET, el nombre del estudiante y la fecha del reporte que desea consultar.

Se tomaron en cuenta las sugerencias de los profesores, éstos sugirieron que se mostrará un reporte que incluyera los siguientes datos:

Nombre del estudiante

Matrícula del estudiante

Fecha de la sesión

Nivel de KET

Grupo del estudiante

Temas evaluados con su respectiva ponderación de errores y aciertos.

Cabe señalar que un estudiante sólo puede tener un reporte por día, sin importar el número de veces que utilice el sistema durante esa fecha.

La interfaz y el reporte se muestran en las Figuras 4.38 y 4.39.

The screenshot shows a dialog box titled "Select a Report". It contains three dropdown menus: "Ket" with the value "3", "Student Name" with the value "Yetrnalez Quintas Ruiz", and "Date of Report" with the value "30/06/2005". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Figura 4.38. Pantalla para seleccionar un reporte.

The screenshot shows a window titled "Individual Student Report". It displays the following information:

- Name:** Yetrnalez Quintas Ruiz
- ID Number:** 20020198
- Group:** 201-B
- Date:** 07/07/2005
- Ket:** 1
- Date of report:** 30/06/2005

Below the information, there is a table with the following columns: Topic, Correct %, and Incorrect %.

Topic	Correct %	Incorrect %
C		
Colors	100	0

At the bottom left of the window, it says "Page 1 of 1".

Figura 4.39. Reporte individual.

4.3.10. Módulo: Help

La ayuda que se implementó para este sistema se encuentra en Inglés debido a que usuarios de habla inglesa serán quienes lo utilicen. La Figura 4.40 muestra un ejemplo de ésta.

New Item

NEW ITEM

- Select a KET level and then select a Topic.
- Write an item name in the space.
- Write a description for this item. This description will be used for the hangman game and will only be used for KET 2 level.
- Write an instruction for this item. This will be used for the "catch a word" game.
- To select an image file press the first *Search* button and then select a file; the image will appear.
- To select a sound file press the second *Search* button and then select a file. If you want to listen to the audio file, press the *Test* button.
- Press the *Show utterances panel* button and new information will be requested. This additional information will be used for the "catch a word" game.
- Write an utterance in the correct space.
- If the utterance is correct click the *Right* option or if the utterance is incorrect click the *Wrong* option and then press the *Add* button to add the utterance associated with this item. The *Reset* button is used to clean the utterance information.
- If you want to modify the utterance information, first select an utterance and then press the *Modify* button. You can change the utterance and its value.
- If you press the *Delete* button the utterance will be eliminated, but first you have to select an utterance.
- When all fields are filled, press the *Ok* button.

Example:

KET: 1
 Topic: Vegetables
 Item: Onions
 Instruction: Describe the picture.
 Utterance:
 It is a fruit (wrong answer)
 It is a vegetable (Right answer)

Press the *Ok* button.

Figura 4.40. Ejemplo de pantalla de ayuda para el sistema de retroalimentación.

Capítulo 5

5. Resultados

Una vez que el sistema EZ-KET se instaló y configuró en la Sala de Autoacceso de la UTM, fue puesto a disposición de los usuarios para comprobar el aprovechamiento de los estudiantes al utilizarlo y de esta forma corroborar la hipótesis planteada. En este capítulo se presentan los resultados obtenidos y las observaciones derivadas de la interacción de los usuarios con el sistema.

Para comprobar la hipótesis se realizaron una serie de pruebas en base al siguiente proceso:

Los alumnos de nivel KET recibieron dos días de clases de repaso de temas selectos.

El maestro evaluó el conocimiento de dicho alumnos mediante un examen tradicional.

Durante los siguientes tres días los alumnos utilizaron el sistema.

Se compararon los resultados de los exámenes tradicionales con los reportes generados por el sistema.

Dentro de cada nivel KET se evaluaron a cinco alumnos. Los reportes generados por EZ-KET se pueden consultar en el anexo B.

A continuación se muestran los resultados obtenidos en el examen aplicado por el profesor y los resultados obtenidos por el sistema. Las celdas que no tiene número (N.A.) indican que el tema no se practicó, dado que los temas se presentan de forma aleatoria:

La Tabla 5.1 presenta los resultados de la evaluación tradicional obtenidos por tres estudiantes. El porcentaje de aciertos que tres estudiantes (Est.) obtuvieron en tres días de sesión con el sistema se muestra en la Tabla 5.2.

Tabla 5.1. Resultados de la evaluación tradicional.

K E T	Estudiantes	TEMAS			
		Vegetales % Aciertos	Casa % Aciertos	Colores % Aciertos	Frutas % Aciertos
1	1	50	80	100	30
	2	0	20	60	30
	3	0	20	60	30
2		Casa		Verbos	
	1	60		0	
	2	100		20	
	3	80		80	
3		Casa		Verbos	
	1	100		0	
	2	40		20	
	3	80		60	

Tabla 5.2. Resultados obtenidos con el sistema.

		TEMAS							
		Colores	Seguir instrucciones	Frutas	Casa	Números	Viajes	Vegetales	
K E T 1	Día 1	Est.1	33.33	0	100	100	N.A	100	58.83
		Est.2	100	0	73.68	53.33	50	0	66.66
		Est.3	0	0	53.33	70	N.A	100	20
	Día 2	Est.1	100	33.33	76.92	93.33	N.A	100	25
		Est.2	N.A	0	64.7	60	100	100	0
		Est.3	60	50	50	60	100	N.A	100
	Día 3	Est.1	100	100	100	93.33	N.A	100	77.77
		Est.2	100	0	100	80	100	100	41.66
		Est.3	50	0	87.5	80	N.A	100	N.A
		Adjetivos	Rutinas	Fechas	Seguir instrucciones	Casa	Verbos		
K E T 2	Día 1	Est 1	0	66.66	N.A	0	64.86	0	
		Est.2	80	75	100	50	72.41	33.33	
		Est.3	50	100	25	N.A	N.A	100	
	Día 2	Est.1	29.41	60	33.33	0	72.72	17.64	
		Est.2	N.A	66.66	80	0	85.18	62.5	
		Est.3	100	72.72	77.77	0	100	26.08	
	Día 3	Est.1	N.A	60	40	0	85.71	21.42	
		Est.2	N.A	0	N.A	0	88.88	N.A	
		Est.3	N.A	40	100	0	N.A	42.85	
		Ropa	Seguir instrucciones	Casa	Trabajos		Verbos		
K E T 3	Día 1	Est 1	N.A	0	33.33	100	0		
		Est.2	N.A	0	37.03	77.41	36.36		
		Est.3	N.A	0	44.44	68.75	33.33		
	Día 2	Est.1	0	0	N.A	100	18.18		
		Est.2	40	0	37.03	100	90		
		Est.3	100	0	55.55	90	100		
	Día 3	Est.1	87.5	50	29.62	100	54.54		
		Est.2	75	0	38.88	100	75		
		Est.3	100	0	55.55	100	75		

5.1. Comparación del desempeño

Una vez presentados los resultados de la evaluación realizada por el profesor y de los reportes generados por EZ-KET, es posible compararlos y determinar, sí después de utilizar el sistema, el alumno mejoró o no su comprensión en el idioma Inglés, esta comparación se muestra a continuación:

Porcentaje de aciertos de cada estudiante a nivel KET 1 en su evaluación mediante examen (E) y la media obtenida a partir del sistema (S).

Tabla 5.3. Comparación de porcentajes de aciertos KET 1.

Estudiantes	Temas							
	Vegetales % Aciertos		Casa % Aciertos		Colores % Aciertos		Frutas % Aciertos	
	E	S	E	S	E	S	E	S
1	50	53.70	80	95.55	100	77.78	30	92.31
2	0	36.11	20	64.44	60	100	30	79.46
3	0	60	20	70	60	55	30	63.61

Como se puede observar en la Tabla 5.3 los alumnos en su mayoría mejoraron su desempeño mediante el uso del sistema EZ-KET. La Figura 5.1 muestra en forma gráfica los resultados obtenidos. Se puede observar que el estudiante 2 tuvo mejoras todos los temas.

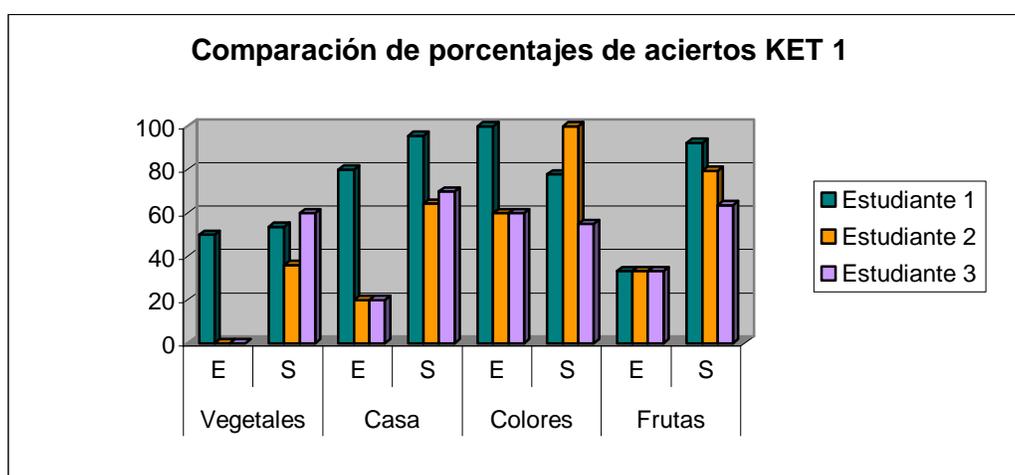


Figura 5.1. Gráfica comparativa de porcentajes de aciertos para KET 1.

Porcentaje de aciertos de cada estudiante a nivel KET 2.

Tabla 5.4. Comparación de porcentajes de aciertos KET 2.

Estudiantes	Temas			
	Casa % Aciertos		Verbos % Aciertos	
	E	S	E	S
1	60	74.43	20	13.02
2	100	82.15	100	47.92
3	80	100	80	56.31

En la Tabla 5.4 se puede ver que los alumnos mejoraron en su mayoría. Sin embargo, mencionaron que algunos de los aspectos evaluados mediante el sistema eran desconocidos para ellos, por ejemplo en el uso de los verbos.

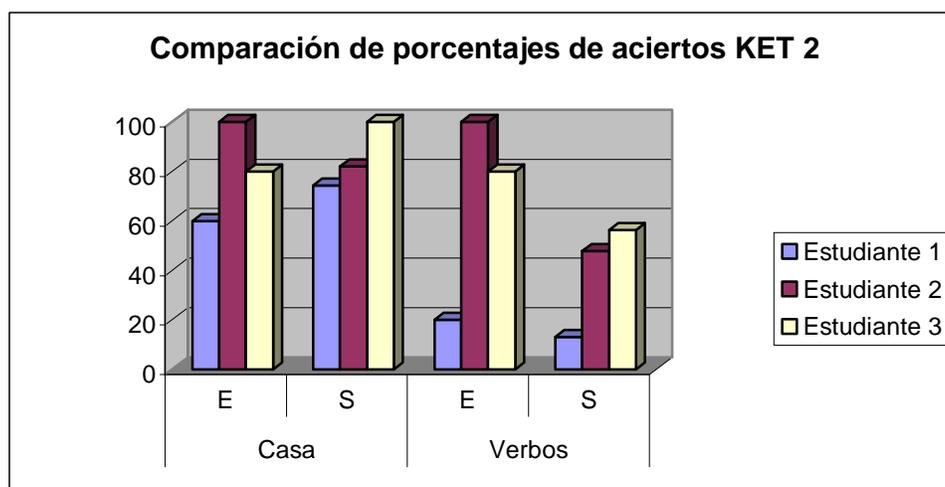


Figura 5.2. Gráfica comparativa de porcentajes de aciertos para KET 2.

La gráfica de la Figura 5.2 muestra que el estudiante 1 tuvo una mejora en el tema “Casa”, pero en el tema “Verbos” no se obtuvo una mejora.

Porcentaje de aciertos de cada estudiante a nivel KET 3 (Tabla 5.5).

Tabla 5.5. Comparación de porcentajes de aciertos KET 3.

Estudiantes	Temas			
	Casa % Aciertos		Verbos % Aciertos	
	E	S	E	S
1	100	31.47	0	24.24
2	40	37.64	20	67.12
3	80	51.84	60	69.44

En este caso, los resultados muestran que en el tema “Casa” ningún estudiante mejoró su desempeño, lo anterior se debe a que el examen tradicional incluyó únicamente tres muebles de la casa, mientras que el sistema EZ-KET evalúa alrededor de 25 muebles por ejercicio, lo que aumentó el grado de dificultad en dicha prueba. En el tema “Verbos” los resultados muestran que en general los estudiantes si mejoraron. La figura 5.3 muestra en forma gráfica estas conclusiones.

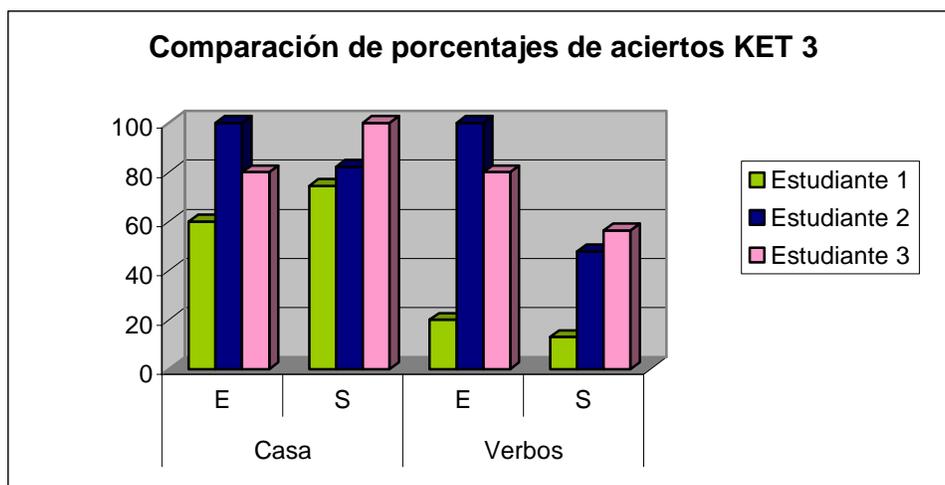


Figura 5.3. Gráfica comparativa de porcentajes de aciertos para KET 3.

Dado lo anterior se puede asegurar que el sistema EZ-KET es realmente una herramienta de apoyo a la enseñanza del idioma Inglés, ya que como mencionaron los mismos usuarios, los motiva a practicar de una manera amena los diferentes temas necesarios para el dominio del vocabulario, gramática y comprensión auditiva en los primeros niveles de enseñanza de este idioma.

Conclusiones y Trabajos Futuros

El software educativo es una parte de las tecnologías de la educación, capaz de transformar contundente y positivamente el proceso de enseñanza-aprendizaje, aportando nuevos y mejores caminos para el desarrollo educativo. Como parte de esa aportación a la enseñanza, EZ-KET ha contribuido a mejorar el dominio del idioma Inglés en los estudiantes de nivel KET de la Universidad Tecnológica de la Mixteca. Comprobar la hipótesis planteada al inicio de este trabajo de tesis, requirió impartir clases de refuerzo, así como la aplicación de exámenes de forma tradicional supervisados por profesores; finalmente el uso continuo del sistema mostró mejoras significativas en los estudiantes y su conocimiento del idioma Inglés.

En cuanto a las teorías del aprendizaje que respaldan el diseño de EZ-KET, es importante puntualizar que no existe una teoría de aprendizaje que sea mejor que otra; sin embargo, existen teorías que se aplican mejor a unos tipos de software que a otros, e incluso, en algunos casos existen más de una teoría que se ajusta a las exigencias del problema a tratar. En este trabajo de tesis se considera el conductismo debido a que se

plantea que el estudiante experimente para aprender: el uso de ejercitadores implica que el estudiante brinde cierta respuesta de acuerdo a los estímulos proporcionados por el sistema. El constructivismo es otra de las teorías del aprendizaje que sustentan este trabajo: el aprendizaje de cada estudiante consiste en la construcción del conocimiento en base a su interacción con el sistema, es decir, su conocimiento se construye cada vez que utiliza el sistema. De acuerdo a las modalidades adoptadas por EZ-KET (ejercitadores y juegos), las teorías mencionadas con anterioridad resultan apropiadas y han brindado buenos resultados.

Como parte del desarrollo de este trabajo de tesis fue necesario adoptar un modelo formal de enseñanza-aprendizaje que EZ-KET debería utilizar. Por tal motivo, este software educativo incorpora el modelo de cuatro etapas de Alessi y Trollip (ver Apartado 1.1.3). Este modelo contribuyó a especificar elementos importantes en la fase de diseño, así como a seleccionar una adecuada presentación de la información dentro del software educativo. También ayudó a identificar una manera adecuada de evaluar el desempeño del estudiante: la generación de reportes cada vez que éste acceda al sistema.

El proceso de desarrollo de EZ-KET presentó diversas fases, presentándose en cada una de ellas una serie de problemas que debían ser resueltos. Durante la fase del análisis, los problemas se relacionaban a la definición y delimitación del sistema, presentándose diferentes opiniones respecto al contenido del software educativo. El contenido fue delimitado con la colaboración directa de los profesores del Centro de Idiomas de la UTM y se procedió a una temprana etapa de selección de juegos y ejercicios. Los estudiantes tuvieron la oportunidad de visualizar los primeros prototipos en papel del sistema, permitiendo señalar lo que les agradaba y lo que no les parecía apropiado en el sistema a desarrollar; los profesores también tuvieron la oportunidad de conocer la propuesta del sistema a través de prototipos realizados para la computadora. Los requisitos del sistema se definieron con base a las opiniones de los profesores y los estudiantes.

En la fase de diseño es sumamente importante señalar que el uso del Diseño Centrado en el Usuario permitió determinar qué pasos seguir para la obtención de un

software de calidad. Sin la presencia de esta metodología, los resultados hubiesen carecido de la calidad necesaria para satisfacer las expectativas planteadas al inicio del proyecto.

La fase de implementación transcurrió sin mayores problemas, esto debido al buen análisis y diseño realizado en fases anteriores. Sin embargo, se presentaron problemas durante la liberación y puesta en marcha del sistema: los equipos disponibles en la Sala de Autoacceso de la UTM carecían de las características apropiadas para que EZ-KET se ejecutase de forma óptima. A pesar de las limitaciones, EZ-KET fue instalado y puesto a prueba. Los resultados fueron satisfactorios, aún con los retrasos propios de la poca capacidad de los equipos, EZ-KET logró sorprender y agradar a la mayoría de los estudiantes que interactuaron con el sistema.

Una vez puesto en marcha, los resultados obtenidos con el uso del sistema EZ-KET mostraron una mejora evidente en las áreas propuestas, logrando así el cumplimiento del objetivo planteado. La diversidad de juegos y ejercitadores, aunados a la capacidad de actualización que ofrece el sistema, hacen de EZ-KET una herramienta atractiva para la práctica autodidacta del idioma inglés. Es apropiado resaltar que los estudiantes que hicieron uso del sistema y que fueron evaluados mediante la técnica tradicional del examen supervisado por profesor mostraron mejoras en casi todos los casos, demostrando una mayor capacidad de respuesta, especialmente en los temas: vegetales, colores, frutas e identificación de las partes de una casa. Se espera que el uso continuo del sistema fortalezca el conocimiento del estudiante, permitiéndole responder cada vez con mayor precisión a las situaciones presentadas por EZ-KET.

Para que EZ-KET siga siendo atractivo al usuario, se proponen como trabajos futuros extender el sistema de monousuario a multiusuario, permitiendo la competencia entre estudiantes en tiempo real; lo cual implica que más de un estudiante participe en los juegos a la vez, aumentando la competitividad y la diversión de los usuarios. También se puede considerar como trabajo futuro la incorporación de nuevos juegos y/o ejercitadores en base a las necesidades y requerimientos que puedan presentarse en el futuro. Igualmente, EZ-KET puede extenderse a otros niveles de enseñanza del idioma Inglés como son el nivel

PET o incluso para la preparación de exámenes avanzados como son *First Certificated* y *TOEFL*. Además, ya que EZ-KET ha demostrado ser una alternativa viable para reforzar el conocimiento del idioma Inglés, es posible pensar en su aplicación en otros idiomas, pudiendo partir del mismo concepto y reutilizando ciertos elementos para la construcción de un nuevo software educativo orientado a otros idiomas. Finalmente, queda como tarea permanente del Centro de Idiomas de la UTM mantener al día la información de este sistema.

Bibliografía

[AC95] Alonso, C., Gallego D., 1995. *Los estilos de aprendizaje. Procedimientos de diagnóstico y mejora*. Ediciones Mensajero, España.

[AD68] Ausubel, D. P., 1968. *Educational psychology: A cognitive view*, Ed. Rinehart and Winston, New York.

[AS00] Alessi, S., Trollip, S., 2000. *Multimedia for learning*, Ed. Allyn & Bacon, Boston.

[BA95] Bork, A., 1995. Why has the computer failed in schools and universities?. *Journal of science education and technology*, Vol. 4, pp. 97-102.

[BB56] Bloom, B., Krathwohl D., 1956. *Taxonomy of educational objectives: Handbok I. The cognitive domain*, Ed. David McKay & Co, New York.

[DA93] Davis, A., 1993. *Software requirements: Objects, functions, and states*, Ed. Prentice Hall, Englewood Cliffs, New Jersey.

[FC97] Faulkner, C., 1997. *The essence of human computer interaction*, Ed. Prentice Hall.

[FX00] Ferré, X., 2000, Principios básicos de usabilidad para ingenieros software. V *Jornadas ingeniería de software y base de datos*. Valladolid, España, pp. 1-8.

[FX03] Ferré, X., 2003, Incrementos de usabilidad al proceso de desarrollo software. VIII *Jornadas ingeniería de software y base de datos*. Alicante, España, pp.1-10.

[GH01] González, H., 2001. *Las métricas de software y su uso en la región*, Universidad de las Américas, México, Tesis de licenciatura.

[GJ02] González, J., 2002. *Integración de un modelo de proceso sistemático en el desarrollo de software educativo*, Universidad de Vigo, España, Tesis de doctorado.

[GL89] García, L., 1989. *La Educación. Teoría y conceptos*, Ed. Paraninfo, Madrid.

[GR02] Glass, R., 2002. *Facts and Fallacies of Software Engineering*, Ed. Pearson.

[GR92] Gangé, R.M., Briggs, L., Wager, W., 1992. *Principles of instructional design*, Ed. Harcourt Brace Jovanovich, New York.

[GT04] Granollers, T., 2004. *Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares*, Universidad de Lleida, España, Tesis de doctorado.

[GT98] Good, T.L., Brophy, J.E., 1998. *Educational psychology: A realistic approach*, White Plains, New York.

[LJ01] Lorés, J., 2001, La interacción persona ordenador. *Congreso Internacional Interacción Persona Ordenador*, España, pp. 1-15.

[LM02] Lozano, M., González, P., 2002. Desarrollo y generación de interfaces de usuario a partir de técnicas de análisis de tareas y casos de uso. *Revista Iberoamericana de Inteligencia Artificial*, No. 16, pp. 83-91.

[LS02] Lawrence S., 2002. *Ingeniería de Software Teoría y Práctica*, Ed. Prentice Hall, Madrid, España.

[LT95] Landauer, T.K., 1995. *The trouble with computers: Usefulness, usability and productivity*, Ed. MIT Press.

- [MD02] Martínez, D., et al. 2002, ¿Es necesaria una ingeniería del software educativo? Constatación de la crisis del software educativo. *IX Congreso Internacional de Investigación en Ciencias Computacionales*. Puebla, México, pp. 1-9.
- [MM91] Merrill, M.D., 1991, Constructivism and instructional design. *Educational technology & Society*, pp.45-43.
- [ND86] Norman, D., Draper, S., 1986. *User Centered System Design*, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ.
- [NJ93] Nielsen, J., 1993. *Usability engineering*, Morgan Kaufmann/Academic Press, USA.
- [NJ95] Nielsen, J., 1995. *Hipertext & Hipermedia*, Academia Press, Ed. United Kingdom, USA.
- [PJ93] Preece, J., 1993. *A guide to usability, human factors in computing*, Addison-Wesley Publishing Company.
- [PR01] Pressman, R., 2001. *Ingeniería de software: un enfoque práctico*, Ed. Mc Graw Hill, España.
- [RJ96] Roschelle, J., Kaput, J., 1996. Educational software architecture and systemic impact: The promise of component software. *Journal of educational computing research*, Vol. 14, No.4, pp. 217-228.
- [RJ98] Richards, J., Rodgers T., 1998. *Enfoques y métodos en la enseñanza de idiomas*, Ed. Cambridge University Press.
- [SB98] Shneiderman, B., 1998. *Designing the User Interface: Strategies for Effective Human Computer-Interaction*, Ed. Addison-Wesley.

[SC95] Stephanidis, C., 1995. Towards User Interfaces for All: Some Critical Issues. Panel Session User Interfaces for All-Everybody, Everywhere, and Anytime. *Symbiosis of Human and Artifact-Future Computing and Design for Human-Computer Interaction, Proc. HCI Intl '95*, Vol. 1, pp. 137-142.

[SD00] Shaffer, D., 2000. *Psicología del desarrollo. Infancia y Adolescencia*, International Thomson Editores, México.

[SI02] Sommerville I., 2002. *Ingeniería de software*, Ed. Pearson Educación, México.

[SI97] Sommerville, I., Sawyer, P., 1997. *Requirements Engineering: A good Practice Guide*, Ed. Wiley.

[SP90] Saettler, P. 1990. *The evolution of American educational technology*, Englewood, CO: Libraries Unlimeted, Inc.

[TA98] Truchard, A. & Katz-Haas, R., 1998. *Usability Interface*, Vol 5, No. 1, pp. 10.

Sitios de Internet

[URL01] Baeza, R., *Arquitectura de la información y usabilidad en la web*, disponible en <http://www.mantruc.com/files/art-epi-2004.pdf>, visitada en febrero 2005.

[URL02] Borges de Barros, H., *Análisis experimental de los criterios de evaluación de usabilidad de aplicaciones multimedia en entornos de educación y formación a distancia*, disponible en http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0716102-102210/04Capitulo04.pdf, visitada en febrero 2005.

[URL03] *Discreet 3D Max*, disponible en <http://www.comgrap.cl/productos/3dmax.htm>, visitada en marzo 2005.

- [URL04] *EDUCARED*, disponible en <http://www.educared.net/aprende/softwareEducativo/index.htm> visitada en noviembre 2004.
- [URL05] Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., Verplank, W., *Curricula for Human-Computer Interaction*. ACM SIGCHI, disponible en: <http://www.acm.org/sigchi/cdg>, 1997, visitada en enero 2005.
- [URL06] Jonassen, D. H., McAleese, T. M. R. *A Manifesto for a Constructivist Approach to Technology in Higher Education*, disponible en <http://led.gcal.ac.uk/clti/papers/TMPaper11.html>, 2002, visitada en noviembre 2004.
- [URL07] Márques, P., *El software educativo*. Disponible en <http://www.didacticahistoria.com/tecedu/tecedu13.htm> visitada en noviembre 2004.
- [URL08] *Microsoft SQL Server*, disponible en <http://www.microsoft.com/latam/sql/evaluation/overview/default.asp> visitada en marzo 2005.
- [URL09] Morales, C., *Modelo de evaluación de software educativo*. Disponible en <http://investigacion.ilce.edu.mx/dice/proyectos/evaluacion/modelo.htm> visitada en noviembre 2004.
- [URL10] *Real Academia Española*, disponible en <http://www.rae.es>, visitada en noviembre 2004.
- [URL11] Sosa J., *Software Educativo para apoyo a la enseñanza de la Técnica de Modelado y Diseño Orientado a Objetos*, <http://www.unne.edu.ar/Web/cyt/com2004/8-Exactas/E-031.pdf>, visitada en agosto 2004.

[URL12] *The Usability Methods Toolbox*, disponible en <http://jthom.best.vwh.net/usability/>, visitada en febrero 2005.

[URL13] Zavala, R. *La Ingeniería de Software*, disponible en <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#IngSoft>, visitada en octubre 2005.

Anexo A. Temario

Temario propuesto por el centro de idiomas de la Universidad Tecnológica de la Mixteca.

Describe people

Tall
Short
Thin
Hair
Eyes
Long
Short
Pretty
Beautiful
Handsome

House

Kitchen
Living room
Bedroom
Bathroom
Dining room
Oven
Fridge
Sink
Armchair
Sofa
Lamp
Coffee table
Curtains
TV
Picture
Bed
Wardrobe
Dressing table
Carpet
Shower
Toilet
Mirror
Towel
Bath
Washbasin
Trash can

Day

Monday

Table
Chairs

Colors

Yellow
Blue
Green
Grey
Pink
Purple
Red
Orange
White
Black
Brown

Fruits

Apple
Orange
Lemon
Strawberry
Melon
Banana
Grapes
Pineapple

Vegetables

Potato
Beans
Carrot
Pepper
Onions
Tomato

Drinks

- Milk
- Coffee
- Water
- Soda

Jobs

Carpenter

Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Nationalities

Mexican
American
British
Spanish
Canadian

Months

January
February
March
April
May
June
July
August
September
October
November
December

Personal belongings

A computer
A car
A bicycle
A calculator
A desk

Plumber
Electrician
Mechanic
Architect
Engineer
Taxi diver
Teacher
Doctor
Waiter
Dentist
Secretary
Student

Clothes

Hat
Shoes
Suit
T-shirt
Shirt
Dress
Skirt
Jeans

Feelings

Happy
Sad
Sick
Fine
Hungry
Thirsty
Hot
Cold

Anexo B. Reportes

Muestra de reportes de los estudiantes a nivel KET 1.

Individual Student Report

Individual Report

Date: 07/09/2005

Name: claudia loyo **Ket:** 1

ID Number: 2004070343 **Date of report:** 06/09/2005

Group: pli-002

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
C		
Colors	33.33	66.66
Follow instructions	0	100
fruits	100	0
House	100	0
Traveling	100	0
Vegetables	58.33	41.66

Page 1 of 1

Individual Student Report

Individual Report

Date: 07/09/2005

Name: claudia loyo **Ket:** 1

ID Number: 2004070343 **Date of report:** 07/09/2005

Group: pli-002

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
C		
Colors	100	0
Follow instructions	33.33	66.66
fruits	76.92	23.07
House	93.33	6.66
Traveling	100	0
Vegetables	25	75

Page 1 of 1

Individual Student Report

Individual Report

Date: 08/09/2005

Name: claudia loyo **Ket:** 1

ID Number: 2004070343 **Date of report:** 08/09/2005

Group: pli-002

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
C		
Colors	100	0
Follow instructions	100	0
fruits	100	0
House	93.33	6.66
Traveling	100	0
Vegetables	77.77	22.22

Page 1 of 1

Muestra de reportes de los estudiantes a nivel KET 2

The screenshot shows a software window titled "Individual Student Report". The window contains the following information:

- Title:** Individual Report
- Date:** 07/09/2005
- Name:** deysyi yolanda rivás palma
- Ket:** 2
- ID Number:** 0101
- Date of report:** 05/09/2005
- Group:** pII_001

Below the student information is a table with three columns: Topic, Correct %, and Incorrect %.

Topic	Correct %	Incorrect %
a		
adjectives	0	100
Daily Routine	66.66	33.33
Follow instructions	0	100
House	64.86	35.13
Verbs	0	100

Page 1 of 1

Individual Student Report

Individual Report

Date: 07/09/2005

Name: deysyi yolanda rivas palma **Ket:** 2

ID Number: 0101 **Date of report:** 06/09/2005

Group: pII_001

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
a		
adjectives	29.41	70.58
Daily Routine	60	40
Dates	33.33	66.66
Follow instructions	0	100
House	72.72	27.27
Verbs	17.64	82.35

Page 1 of 1

Individual Student Report

Individual Report

Date: 07/09/2005

Name: deysyi yolanda rivas palma **Ket:** 2

ID Number: 0101 **Date of report:** 07/09/2005

Group: pII_001

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
D		
Daily Routine	60	40
Dates	40	60
Follow instructions	0	100
House	85.71	14.28
Verbs	21.42	78.57

Page 1 of 1

Muestra de reportes de los estudiantes a nivel KET 3

The screenshot shows a window titled "Individual Student Report" with a standard Windows-style title bar and menu bar. The main content area displays the following information:

Individual Report

Date: 07/09/2005

Name: Adolfo Lopez **Ket:** 3

ID Number: 1985 **Date of report:** 05/09/2005

Group: PLI001

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
F		
Follow instructions	0	100
House	33.33	66.66
Jobs	100	0
Verbs	0	100

Page 1 of 1

Individual Student Report

Individual Report

Date: 07/09/2005

Name: Adolfo Lopez **Ket:** 3

ID Number: 1985 **Date of report:** 06/09/2005

Group: PLI001

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
C		
Clothes	0	100
Follow instructions	0	100
Jobs	100	0
Verbs	18.18	81.81

Page 1 of 1

Individual Student Report

Individual Report

Date: 07/09/2005

Name: Adolfo Lopez **Ket:** 3

ID Number: 1985 **Date of report:** 07/09/2005

Group: PLI001

<u>Topic</u>	<u>Correct %</u>	<u>Incorrect %</u>
C		
Clothes	87.5	12.5
Follow instructions	50	50
House	29.62	70.37
Jobs	100	0
Verbs	54.54	45.45

Page 1 of 1

Anexo C. Manual de usuario

Introducción

Este documento tiene por objeto auxiliar al usuario en el uso del sistema EZ-KET, así como especificar la información que necesita cada módulo e indicar la forma en que se debe utilizar cada juego para lograr los resultados esperados.

Registro de Usuario

Los datos que se deben de proporcionar son los siguientes:

ID Number: Se debe proporcionar el número de matrícula.

Password: Se debe escribir una contraseña (máximo de caracteres 8).

Retype Password: Se vuelve a escribir la contraseña para rectificarla.

Name: Se debe escribir el nombre del estudiante.

Last Name: Escribe los apellidos del estudiante.

KET Level: Selecciona un nivel de KET.

Group: Escribe el grupo en el que te encuentras. Ejemplo: 101-A

Birthday: Se debe proporcionar la fecha de nacimiento. Seleccionar un mes, el día (dos caracteres) y el año (cuatro caracteres).

Una vez proporcionada la información se debe presionar el botón *Submit* para que la información se almacene. Si se desea borrar la información se debe presionar el botón *Reset*. La Figura 1 muestra este módulo.

Figura 1. Registro de usuarios.

Recordar contraseña

La información que se debe proporcionar para que el sistema devuelva la contraseña olvidada por el usuario es:

ID Number: Se debe proporcionar el número de matrícula.

Birthday: Se debe proporcionar la fecha de nacimiento. Seleccionar un mes, el día (dos caracteres) y el año (cuatro caracteres).

Para continuar con el proceso se debe presionar el botón *Ok* y la contraseña aparecerá en el recuadro *Password*. Se debe cerrar la ventana. La interfaz correspondiente a este módulo aparece en la Figura 2.

Figura 2. Recordar contraseña.

Cambiar información

Sólo se puede cambiar el nivel KET y/o el grupo del usuario. Para poder realizar estos cambios se debe proporcionar la siguiente información:

ID Number: número de matrícula.

Password: contraseña.

Se debe presionar el botón *Search* para que el sistema busque la información correspondiente. La información del nivel de KET y el grupo aparecerán en la parte inferior de la ventana.. Se selecciona un nuevo nivel de KET y/o se escribe el nuevo grupo.

Para que la información se guarde se presiona el botón *Ok*. En caso de que no se quiera guardar la información se debe presionar el botón *Cancel*. En la imagen 3 se puede observar la interfaz correspondiente a este módulo.

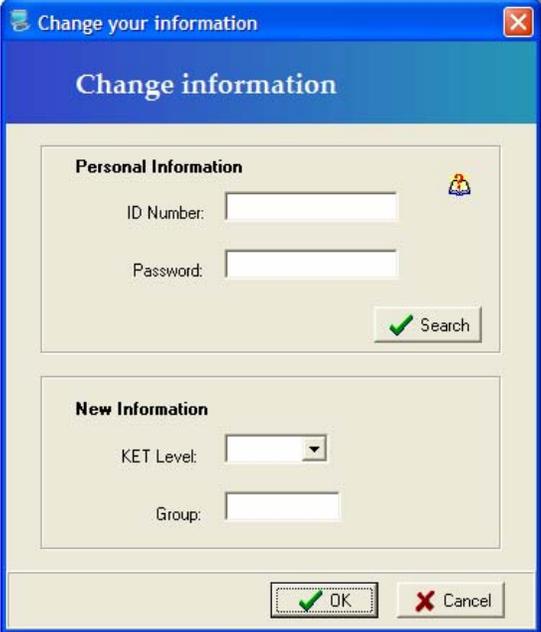


Figura 3. Módulo para cambiar nivel de KET y grupo.

Iniciar sesión

Para poder jugar, el alumno tiene que ser un usuario registrado. La información que el sistema necesita para poder entrar al juego es la siguiente:

ID Number: Se debe proporcionar el número de matrícula.

Password: Se debe escribir la contraseña.

Una vez proporcionada esta información se debe presionar el botón *Sign in* y después aparecerá la ventana que tiene el juego *Serpientes y Escaleras* (Figura 5) para empezar a jugar se debe presionar el botón *Roll dice*.

En la ventana *inicio de sesión* aparecen otras opciones las cuales se muestran a continuación:

Forgotten your password: Esta opción sirve para recordar la contraseña.

Change information: Permite cambiar el nivel de KET y/o el grupo.

If you are new, sign up now: Permite registrarte al sistema, si todavía no lo has hecho.

En la Figura 4 se puede observar dicho módulo.

The image shows a 'Sign In' window with a blue title bar. Inside, there are two text input fields labeled 'ID Number:' and 'Password:'. Below the password field is a button with a green checkmark and the text 'Sign in'. Underneath the button are three links: 'Forgotten your password?', 'Change information', and 'If you are new, Sign Up Now'. A small icon of a person is located at the bottom right of the window.

Figura 4. Módulo para iniciar sesión.

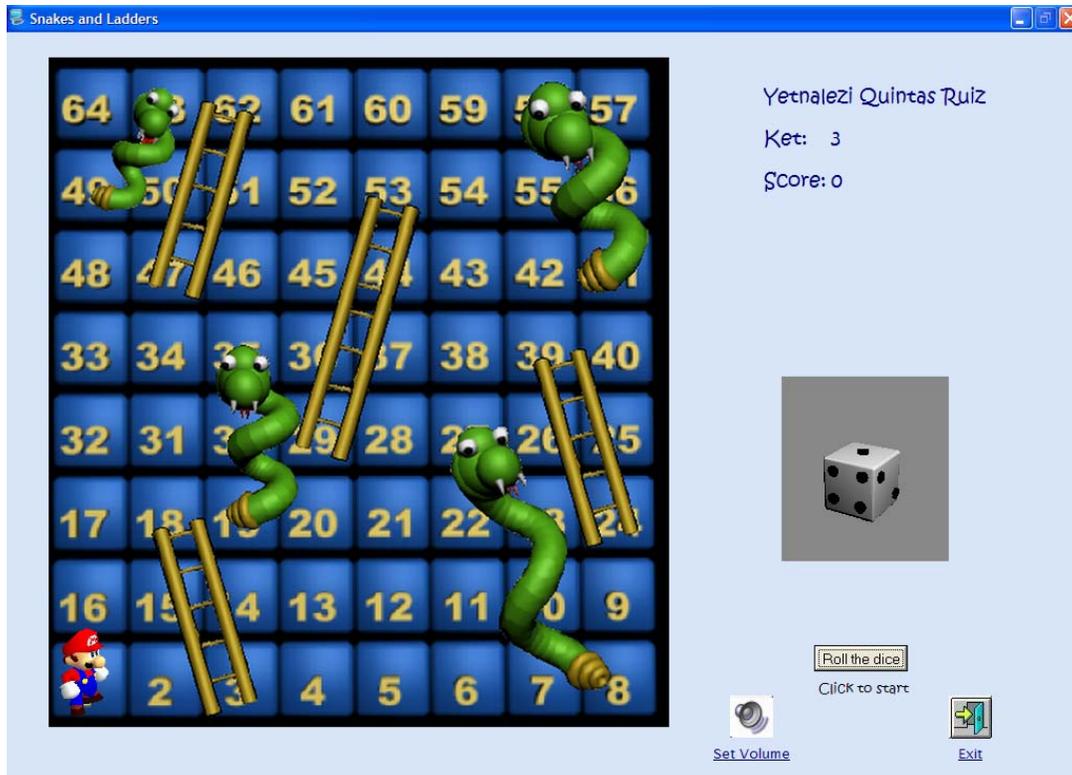


Figura 5. Juego serpientes y escaleras para KET 3.

Juego: Ahorcado

El juego consiste en adivinar la palabra correspondiente a un tema en base a: la imagen (KET 1), la descripción (KET 2) o el sonido descriptivo de la palabra (KET 3) que se encuentra del lado derecho. Si se desea escuchar la pronunciación de dicha palabra (nivel KET 1 y KET 2) se debe presionar el botón de la imagen de una bocina.

En la parte inferior de la ventana se muestra un abecedario. Se debe presionar una letra. Si la letra que se presiona se encuentra dentro de la palabra, ésta aparecerá en la línea. En caso contrario aparecerá una imagen de una parte del cuerpo humano del lado izquierdo. Si no se adivina la palabra, entonces se habrá perdido.

Ejemplo KET 1: Aparece una imagen en el lado derecho, es el color negro, por lo tanto la palabra a adivinar es “Black”. Se deben presionar las letras que aparecen en la parte inferior, esto es: la “B”, “L” y así sucesivamente hasta terminar la palabra. En la Figura 6 se observa el juego para este nivel.



Figura 6. Juego del ahorcado para KET 1.

Ejemplo KET 2: En el lado derecho de la pantalla se muestra una descripción de la palabra a adivinar, en este caso *“It’s a verb that you do in the morning”*, por lo tanto la palabra a adivinar es *“Get up”*. Se deben presionar las letras que aparecen en la parte inferior, es decir se debe presionar la *“G”*, *“E”* y así sucesivamente hasta terminar la palabra. En la Figura 7 se observa el juego para este nivel.



Figura 7. Juego del ahorcado para KET 2.

Ejemplo KET 3: En el lado derecho se muestra un botón que tiene la imagen de una bocina, se debe presionar para saber cuál es la palabra a adivinar, en este caso es “Engineer”. Por lo tanto, se deben presionar las letras que aparecen en la parte inferior, es decir se debe presionar la “E”, “N” y así sucesivamente hasta terminar la palabra. En la Figura 8 se observa el juego para este nivel



Figura 8. Juego del ahorcado para KET 3.

Juego: Relacionar imágenes con palabras

Este juego consiste en arrastrar las palabras que se presentan en el lado derecho de la ventana hacia las imágenes que se muestran en el lado izquierdo. Si se desea escuchar el sonido correspondiente a cada palabra, se debe presionar el botón que tiene la imagen de una bocina.

Si la palabra no concuerda con la imagen, ésta aparece de color rojo, en caso contrario aparece en color verde. Esto quiere decir que el color verde indica correcto y el color rojo incorrecto.

En la parte superior de la ventana aparece el nombre del tema referente a las imágenes. Esto es para todos los niveles KET.

Ejemplo: En el lado derecho aparecen las palabras “Blue”, “Black”, “Red” y “Brown” y del lado izquierdo se encuentran los colores. Lo que se tiene que hacer es arrastrar la palabra “Blue” hasta la imagen correspondiente al color azul y así sucesivamente hasta terminar todas las palabras. Al final se muestra en color rojo a aquellas palabras que estuvieron mal y en color verde a aquellas que estuvieron bien. En la Figura 9 se muestra este juego.

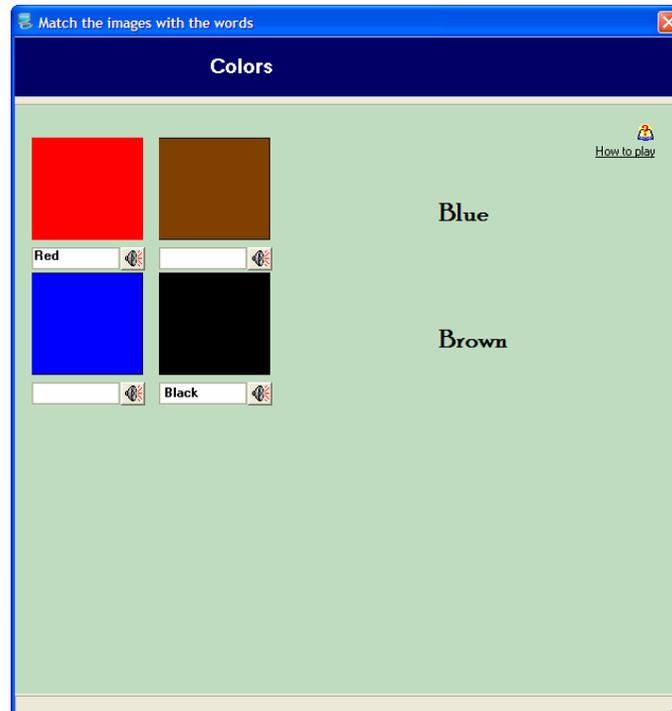


Figura 9. Juego de relacionar imágenes con palabras.

Juego: Cachar palabras

Este juego consiste en cachar las palabras relacionadas con la imagen que aparece en el lado izquierdo de la ventana. Para empezar a jugar se debe de presionar el botón que dice *Play*.

Con el carrito de supermercado se atrapan las palabras, para poderlo mover se usa el ratón (mouse). La instrucción aparece en la parte superior de la ventana. Si la palabra atrapada es incorrecta, ésta se escribe en el rectángulo del lado derecho de lo contrario aparece en el rectángulo del lado izquierdo. Por cada palabra correcta se obtienen 100

puntos y por cada palabra incorrecta se restan 100 puntos. Si no se atrapan palabras no se suman puntos.

Ejemplo: Se debe de presionar el botón *Play*. En el lado izquierdo aparecerá la imagen de un maestro, lo que se tiene que hacer es atrapar las palabras que estén relacionadas con esta imagen. Por ejemplo: “He works in the school” y “He teaches students”. Si éstas se atrapan, aparecerán en el recuadro que dice “right answer”, en caso contrario aparecerán en el que dice “wrong answer”. En la Figura 10 se puede observar este juego.

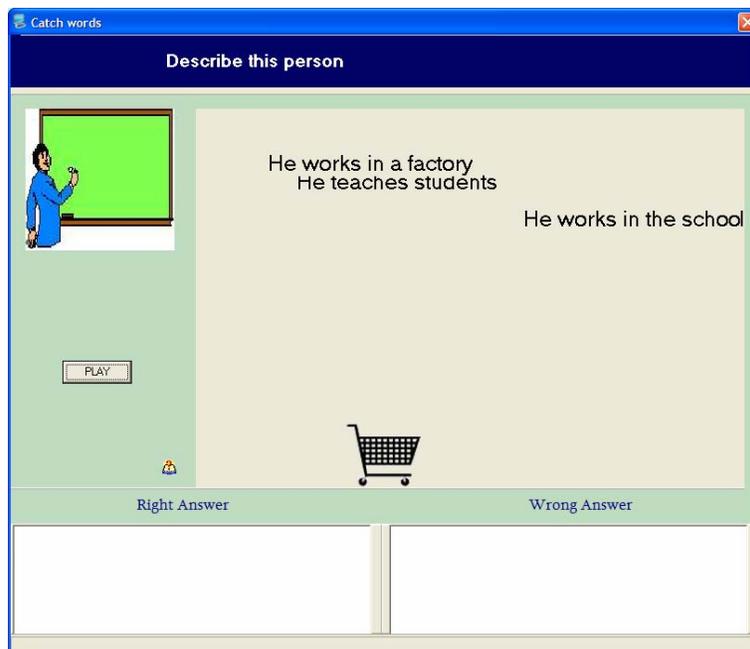


Figura 10. Juego: Cachar palabras.

Juego: Mapa

Este juego consiste en seguir instrucciones, las cuales aparecen en pantalla (sólo para nivel KET 1 y KET 2) y al final decir a que lugar se llegó. Se puede escuchar la instrucción presionando el botón que tiene la imagen de una bocina. Aparece la imagen de una persona ubicada en el mapa, ese es el punto de partida. Presionando las flechas (que aparecen alrededor de la imagen de la persona), ésta se desplaza.

Ejemplo KET 1 o KET 2: La instrucción a seguir aparece en la parte superior. Si se desea escuchar la instrucción se debe presionar la bocina. El punto de partida es “Jamison Hotel”, ya que en ese lugar aparece la imagen de la persona. Se presiona la flecha para poder desplazar a la persona. Una vez realizada la instrucción se debe seleccionar cuál es el lugar al que se llegó.

El juego del mapa correspondiente a los niveles de KET 1 y KET 2 aparece en la Figura 11.

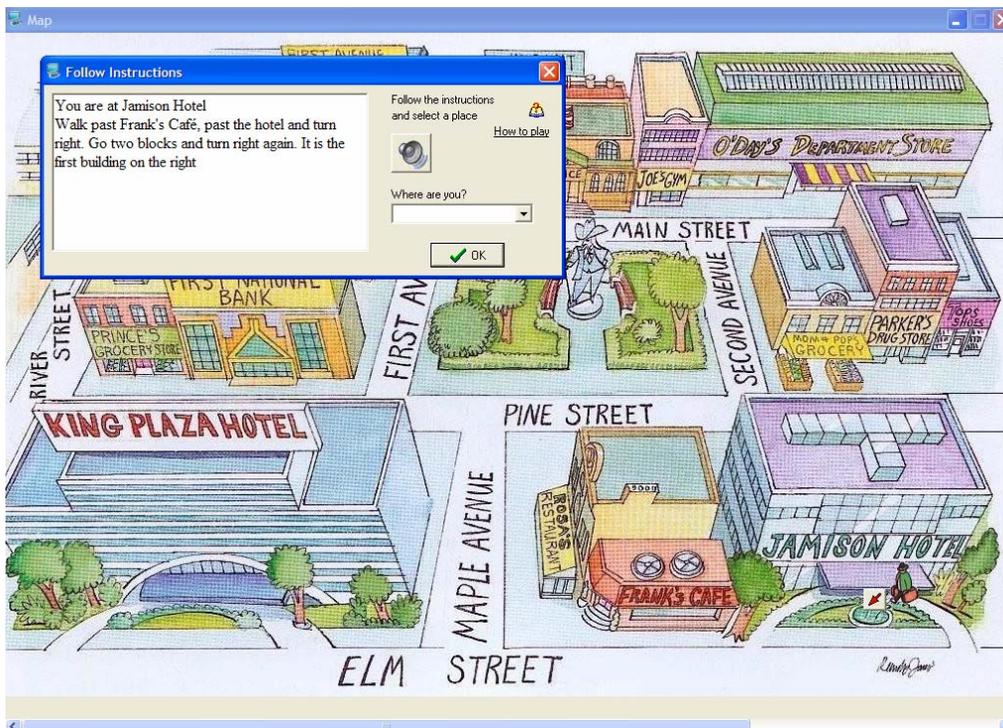


Figura 11. Juego del mapa para los nivel KET 1 y KET 2.

Ejemplo KET 3: Se debe presionar la bocina para poder escuchar la instrucción. Ésta es la única forma en que se presenta la instrucción a seguir. El punto de partida es “Laundromat”, ya que en ese lugar aparece la imagen de la persona. Se deben presionar las flechas para poder desplazarse. Una vez realizada la instrucción se debe seleccionar cuál es el lugar al que se llegó. El juego del mapa correspondiente al nivel de KET 3 aparece en la Figura 12.



Figura 12. Juego del mapa para el nivel KET 3.

Juego: Jugando con los Verbos

Se debe presionar cualquier botón verde. Estos botones indican el número que determina la complejidad del verbo. En la parte superior se especifica la letra con la que empieza el verbo.

Una vez presionado el botón aparece la pregunta. En la parte superior se encuentra el valor de dicha pregunta. En donde dice *Your answer* debes escribir la posible respuesta, y después presionar el botón *Check your answer*. Sólo si la respuesta es correcta sumas los puntos, en caso de ser incorrecta se muestra la respuesta correcta. Para poder terminar este juego debes reunir 1000 puntos.

Ejemplo: Se presiona el botón que tiene el número 400, de la columna que dice “Verbs starting with D and E”. A continuación aparece otra ventana (figura 9), la cual tiene la siguiente pregunta: “The workers ___ a hole next to the street. Past tense of dig”. La respuesta se debe escribir donde dice *Your answer*. En este caso, se escribe *dug* y después

se presiona el botón *Check your answer*. Como la respuesta es correcta aparece un mensaje que dice *right answer* y los puntos se suman. Este juego aparece en las Figuras 13 y 14.



Figura 13. Juego de verbos.

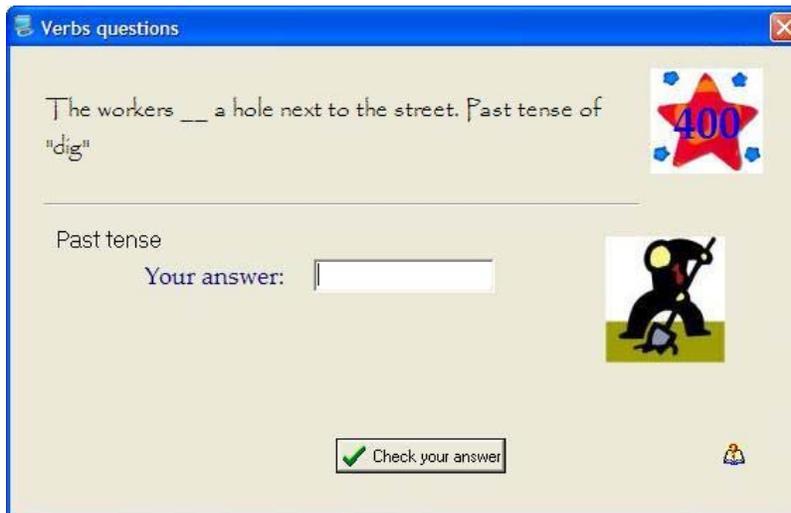


Figura 14. Ventana en donde se realiza la pregunta en el juego de verbos.

Juego: Describe, localiza partes y objetos de una casa

Este juego consiste en identificar las partes y/o muebles de una casa. Se muestra una imagen referente a dicho elemento del lado izquierdo de la pantalla y del lado derecho aparecen los nombres de cada uno de ellos. Se debe de seleccionar (KET 1), arrastrar (KET 2) ó escribir (KET 3) uno de los nombres dependiendo de la imagen que aparece y después

presionar el botón *Next* y así sucesivamente hasta terminar. Al final aparece una ventana indicando los errores y los aciertos obtenidos. Si se desea escuchar la pronunciación de cada elemento de la casa se debe presionar el botón que tiene la imagen de una bocina.

Ejemplo KET 1: La imagen que aparece del lado izquierdo corresponde a la sala de una casa, por tal motivo, se debe seleccionar la opción que dice “Living room” y después presionar el botón *Next*. En la Figura 15 se observa la interfaz para este juego.

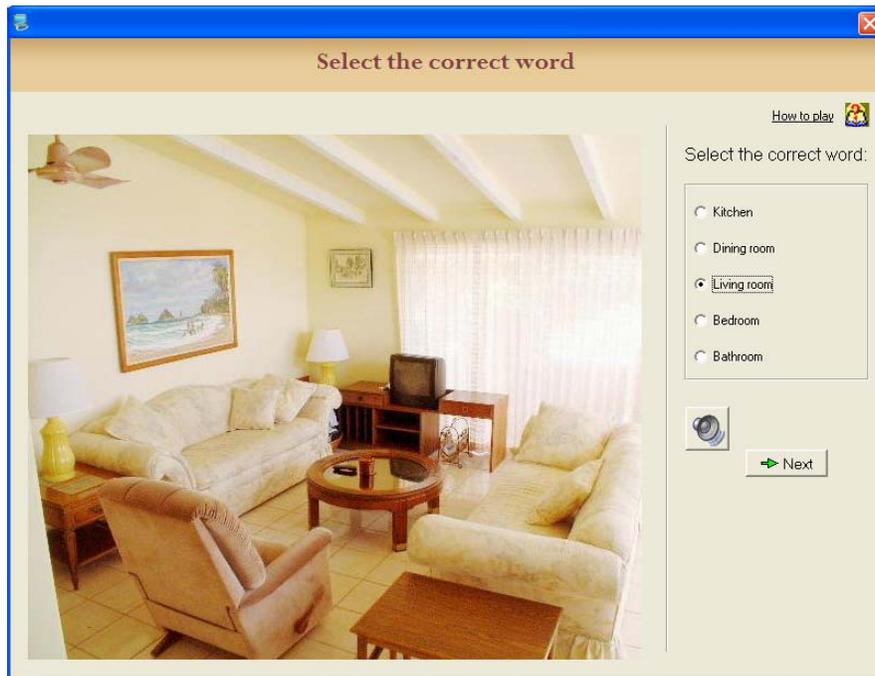


Figura 15. Juego de la casa para KET 1.

Ejemplo KET 2: Las palabras que aparecen en el lado derecho son “Lamp”, “TV”, “Sofa”, “Coffee table”, “Armchair”, “Picture”, “Curtains”, se debe arrastrar la palabra “Lamp” a la imagen de la lámpara, la palabra “TV” a la imagen de la televisión y así sucesivamente hasta terminar con todas las palabras, después se presiona el botón *Next*. En la Figura 16 se observa la interfaz de este juego

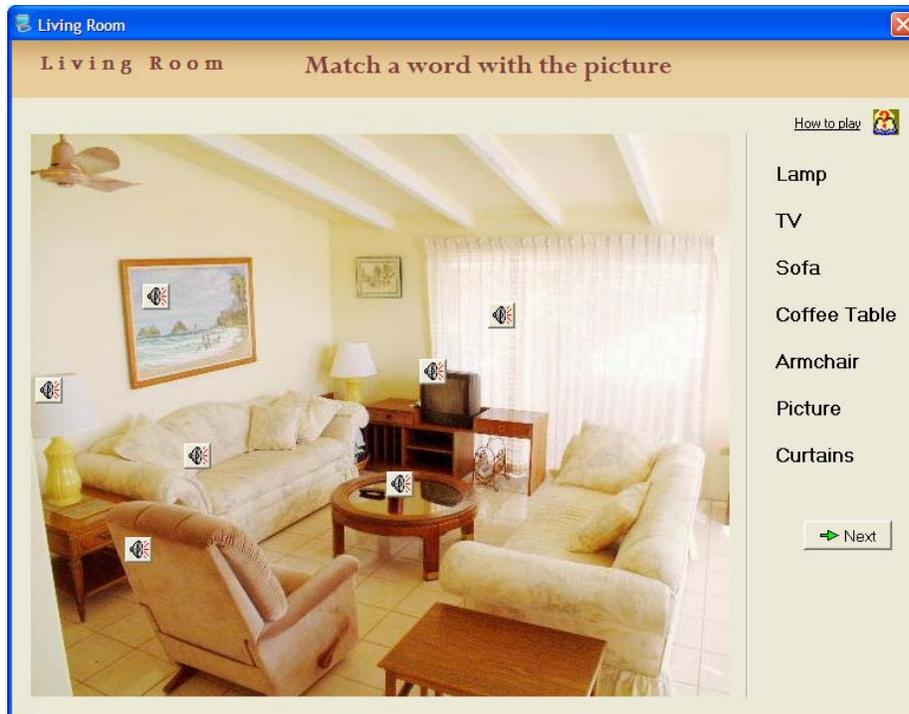


Figura 16. Juego de la casa para KET 2.

Ejemplo KET 3: Aparecen recuadros vacíos en el lado derecho, con una letra indicativa y sobre los muebles se encuentran dichas letras, lo que se tiene que hacer es escribir el nombre del mueble asociado a la letra. La letra A indica la tarja lo que obliga a escribir “sink” en el recuadro de la letra A. En el recuadro de la letra E se debe de escribir “microwave” y así sucesivamente hasta terminar con todos los muebles etiquetados, después se presiona el botón *Next*. En la Figura 17 se observa la interfaz de este juego.

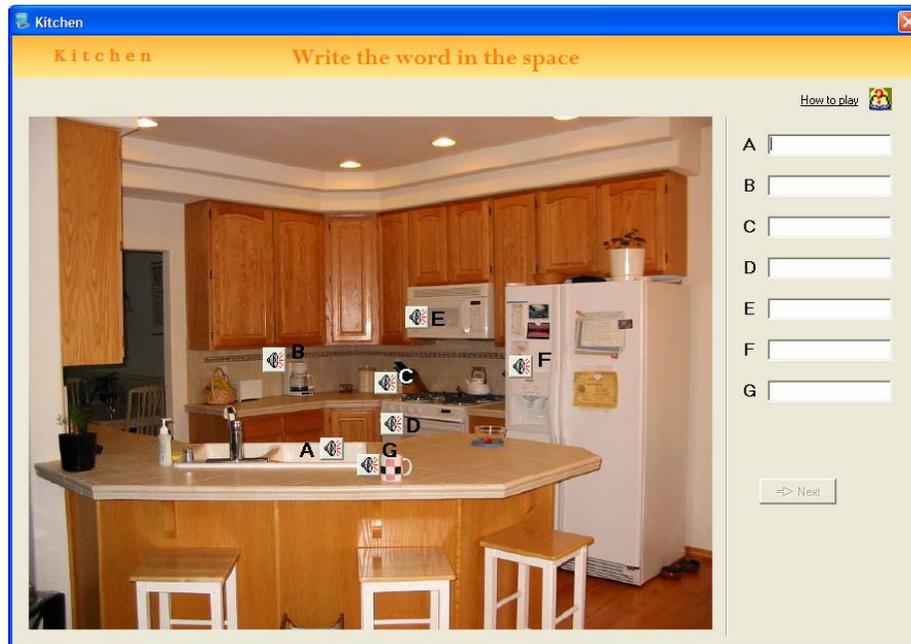


Figura 17. Juego de la casa para KET 3.

Ejercicio tipo A

Este ejercitador consiste en seleccionar una imagen de tres que se presentan. Un enunciado aparece en la parte superior de la ventana. Una vez seleccionada la opción se debe presionar el botón *Ok* y a continuación aparecerá una ventana indicando si la respuesta fue correcta o incorrecta.

Ejemplo: El enunciado es “It’s monkey’s favorite food”, las imágenes que aparecen son: una manzana, uvas y unos plátanos. La imagen correcta a seleccionar son los plátanos, después de presionar el botón *Ok*, aparece una ventana que dice “That’s the correct answer!”. La Figura 18 muestra la interfaz de este tipo de ejercicio.

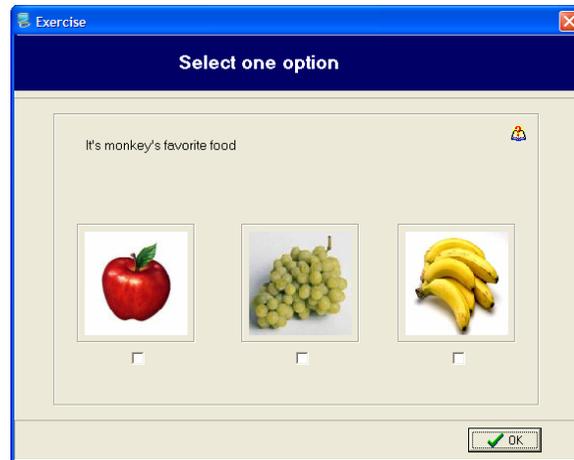


Figura 18. Ejercicio tipo A.

Ejercicio tipo B

Este ejercitador consiste en seleccionar una opción de las cuatro que se presentan. Una pregunta aparece en la parte superior de la ventana. Una vez seleccionada la opción se debe presionar el botón *Ok*. A continuación aparece una ventana indicando si la respuesta fue correcta o incorrecta.

Ejemplo: La pregunta es “What color is a carrot?”, las opciones que aparecen son “purple”, “green”, “orange” y “yellow”. La opción que se debe seleccionar es “orange”. Después de presionar el botón *Ok* aparece una ventana que dice “That's the correct answer!”. La Figura 19 muestra la interfaz de este tipo de ejercicio.

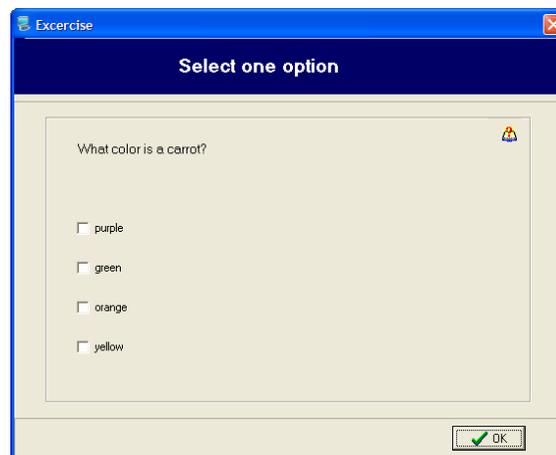


Figura 19. Ejercicio tipo B.

Anexo D. Manual de instalación

Introducción

El presente documento tiene por objeto la ejemplificación de los pasos a seguir para lograr la instalación exitosa del sistema EZ-KET, de forma que éste se ejecute sin contratiempos.

A lo largo de este documento encontrará aspectos relacionados con la instalación del sistema y operaciones de mantenimiento posteriores que tienen como finalidad restaurar la base de datos después de que se ha presentado un problema de pérdida de información.

A grandes rasgos, el proceso de instalación está formado de cuatro actividades básicas:

Instalación y configuración del servidor de bases de datos.

Instalación del sistema EZ-KET.

Configuración de dispositivos y restauración de la base de datos.

Configuración de los equipos clientes.

En las siguientes secciones se mostrarán a detalle cada una de estas actividades. Antes de eso, se presentarán las características que se requieren para que el sistema funcione con normalidad.

Requisitos mínimos

Antes de continuar asegúrese de que su equipo de cómputo reúne las características necesarias para ejecutar adecuadamente el sistema EZ-KET.

Los requisitos mínimos que solicita el sistema se muestran a continuación:

Sistema Operativo Windows 2000 o superior (requisito indispensable del servidor, los equipos clientes pueden utilizar Windows 98 o Me).

64 MB de RAM.

200 MB de espacio en disco duro.

Conexión a red local (opcional).

Tarjeta de sonido (opcional)

En el caso de la conexión a red local puede omitirse si se desea instalar únicamente en un solo equipo, el cual realizará las funciones de servidor y cliente.

Por otro lado, se puede prescindir de la tarjeta de sonido y el sistema todavía podrá funcionar, sin embargo, algunos de los juegos del sistema no darán los resultados esperados debido a la ausencia de sonido.

Instalación y configuración del servidor de bases de datos

En este sistema, el manejador de bases de datos utilizado es SQL Server 2000, el cuál proporciona flexibilidad y robustez a la aplicación. Para la correcta instalación del SQL Server 2000 consulte los manuales que se incluyen con los discos de instalación.

En lo que respecta a la configuración del servidor de bases de datos es importante que considere usted lo siguiente: el sistema EZ-KET se conecta a la base de datos mediante una autenticación de usuario, por lo cual es muy importante que al momento en que el instalador del SQL Server le solicite la clave para el usuario *sa* usted ingrese la palabra *utm*, la cual funciona como password de acceso total al manejador de la base de datos.

Una vez concluida la instalación y configuración del SQL Server 2000, el servicio deberá estar ejecutándose en el equipo servidor, por lo cual aparecerá un icono como el que se muestra en la Figura 1 en el área de notificación del sistema operativo Windows que esté utilizando.



Figura 1. Servicio SQL Server 2000 ejecutándose.

En algunas ocasiones, el Servicio SQL Server 2000 requiere que se reinicie el equipo para que pueda activarse.

Instalación del sistema EZ-KET

Para la instalación del sistema EZ-KET debe copiar la carpeta BDE que se encuentra en el disco de instalación del sistema EZ-KET a la ruta que usted decida que albergará al sistema. Una vez que haya realizado eso, cree un acceso directo en el escritorio que apunte al archivo *EZProject.exe* que se encuentra en la carpeta BDE que acaba de copiar. Repita el proceso anterior para cada equipo donde se ejecutará el sistema.

Cuando haya terminado de instalar el sistema en cada equipo de la red, ubique el archivo *Servidor.txt* que se encuentra en la carpeta BDE del servidor. Modifique su contenido, el archivo deberá contener el nombre del equipo con formato de acceso a recurso de red. Por ejemplo, si el servidor se llama **CA16**, el archivo deberá contener la cadena `\\CA16\`. **No repita este paso en los equipos clientes.**

Configuración de dispositivos y restauración de la base de datos

Una vez instalado y configurado el servidor de bases de datos es necesario configurar los dispositivos y cargar la información básica para que el sistema funcione.

Para la configuración de los dispositivos debe ejecutar el archivo *Config_dispositivos.sql* en el analizador de consultas del SQL Server 2000.

Cuando haya realizado el paso anterior, copie el archivo ResDB.bak que se encuentra en el disco de instalación del sistema EZ-KET a la ruta C:\. Si no copia el archivo la instalación no podrá continuar.

El paso a seguir consiste de restaurar la base de datos, para lo cual es necesario ejecutar el archivo *restaura_db.sql* en el analizador de consultas del SQL Server 2000.

Cuando haya terminado el paso anterior ya tendrá instalada la base de datos en el equipo servidor. **No repita la instalación de la base de datos en los equipos clientes.**

Configuración de los equipos clientes

La configuración de los equipos clientes requiere forzosamente de la creación de un origen de datos ODBC para uso de la aplicación. Entre a *Panel de control* y ahí abra *Herramientas administrativas* (Figura 2).

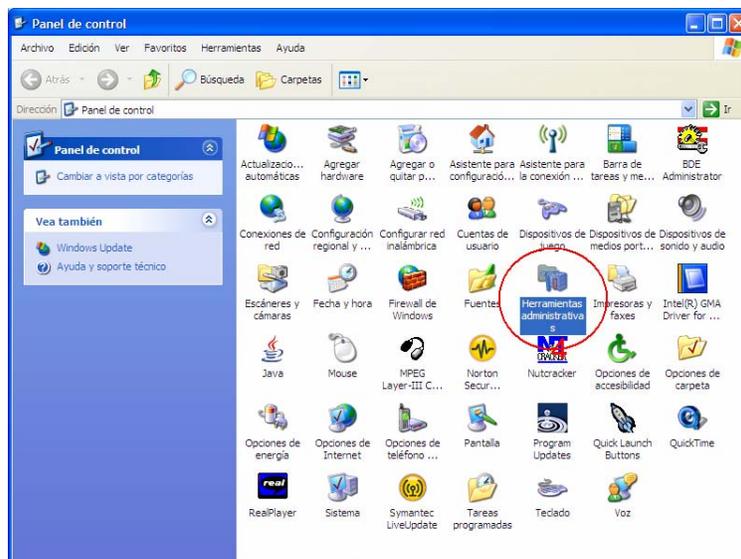


Figura 2. Ubicación de herramientas administrativas dentro del Panel de control.

Ya en *Herramientas administrativas*, abra *Orígenes de datos (ODBC)*. Esto se muestra en la Figura 3.

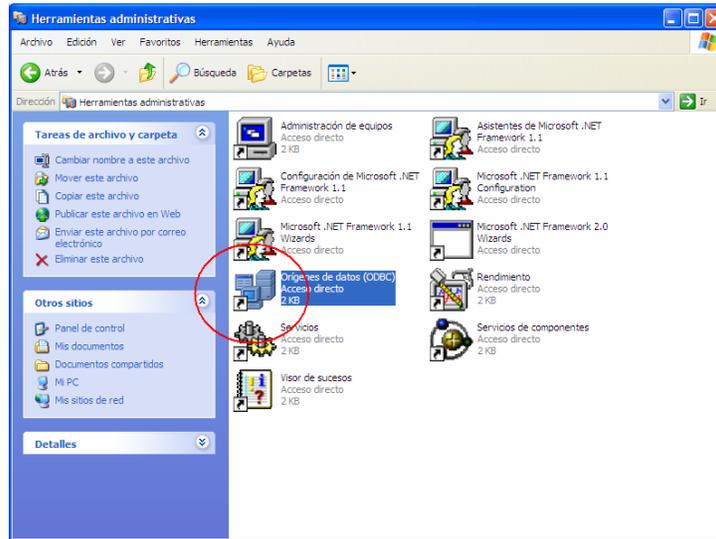


Figura 3. Ubicación de Orígenes de datos (ODBC) dentro de Herramientas administrativas.

Cuando haya abierto la carpeta *Orígenes de datos (ODBC)*, presione el botón *Agregar* (Figura 4).

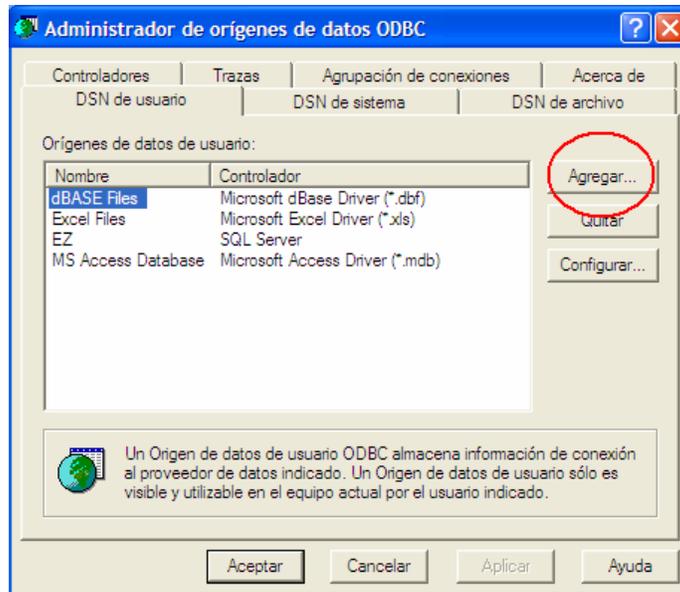


Figura 4. Administrador de orígenes de datos ODBC.

Ahora debe seleccionar el controlador para el nuevo origen de datos. Seleccione SQL Server de la lista y presione **Finalizar** (Figura 5).

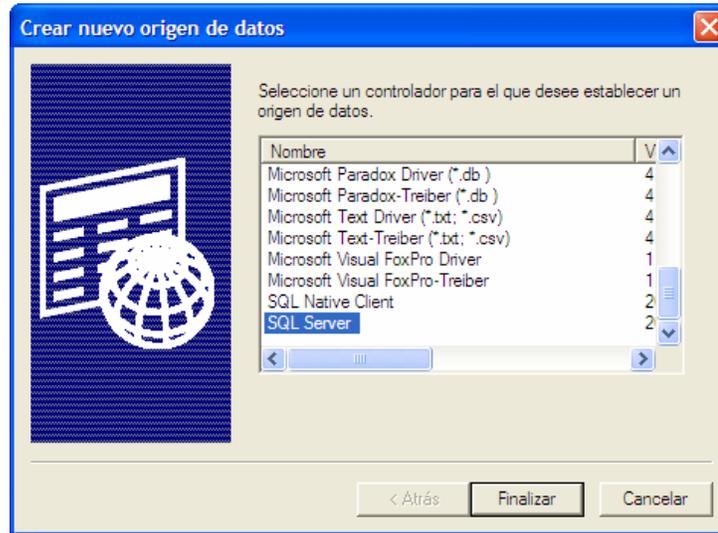


Figura 5. Creación de un nuevo origen de datos.

Especifique el nombre del nuevo origen de datos; este nombre deberá ser **EZ**, de lo contrario el sistema no será capaz de encontrarlo al ejecutarse. También debe especificar el nombre del servidor. Continuando con el ejemplo propuesto, si el servidor se llamase **CA16** este nombre debería aparecer en este campo (Figura 6).

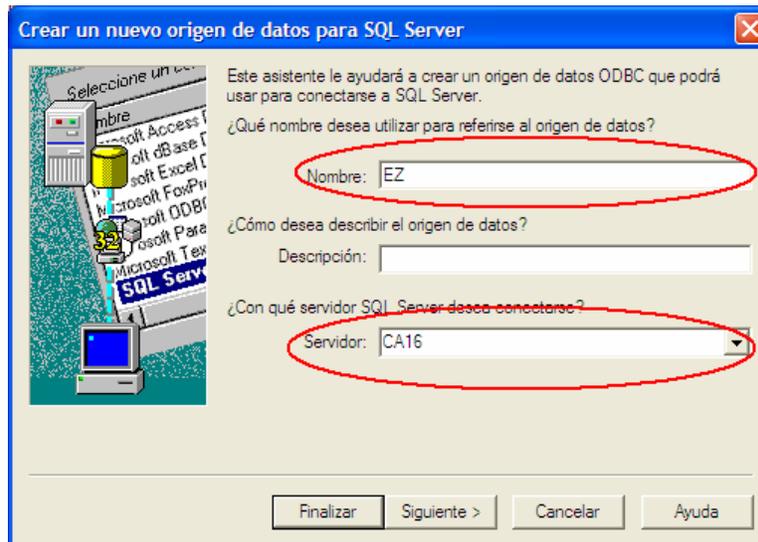


Figura 6. Configuración del nombre y servidor del origen de datos.

En este punto debe configurar el nombre de usuario y la contraseña. Vea la Figura 7 para mayor detalle acerca de la configuración de este paso.

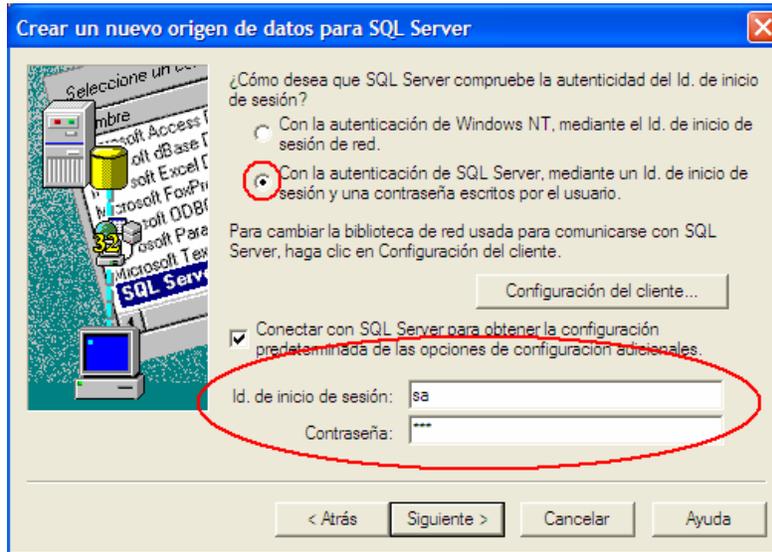


Figura 7. Configuración del nombre de usuario y contraseña.

En la siguiente ventana, habilite la opción que se muestra en la Figura 8, seleccione de la lista desplegable la base de datos **EZ**. Acto seguido presione el botón **Siguiente**.

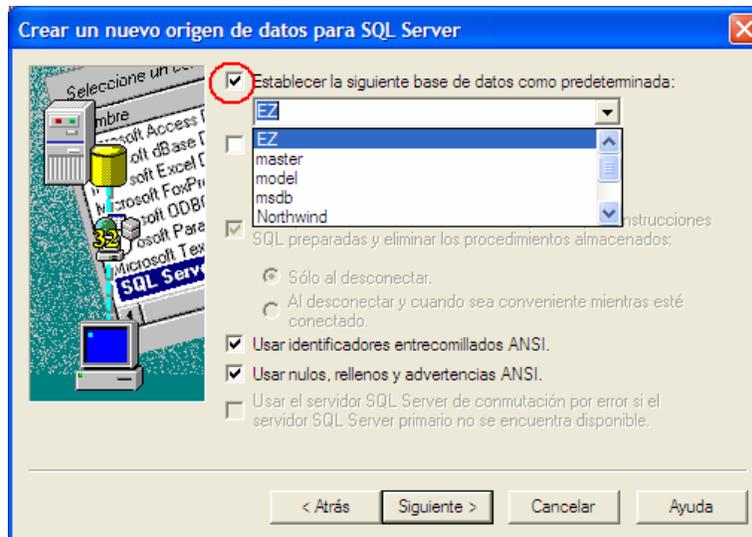


Figura 8. Selección de la base de datos predeterminada.

Presione el botón **Siguiente** hasta que aparezca la ventana de la Figura 9, en la cual debe presionar el botón **Probar** origen de datos.

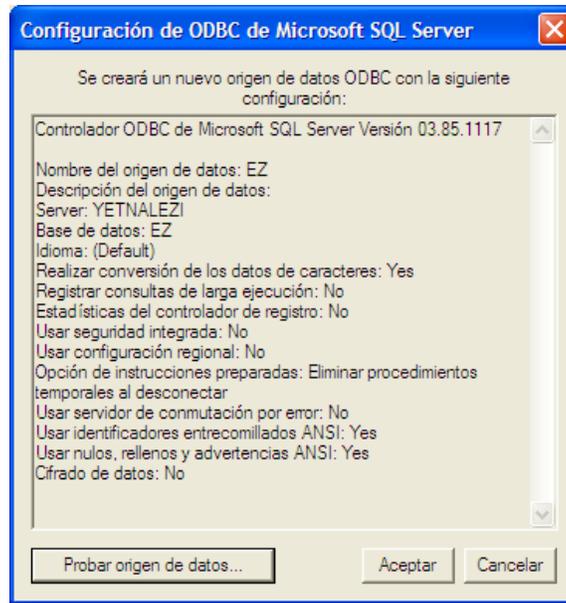


Figura 9. Prueba del origen de datos.

Si la prueba es exitosa, la ventana deberá verse como la de la Figura 4.10. De lo contrario, se ha presentado algún error en la comunicación.

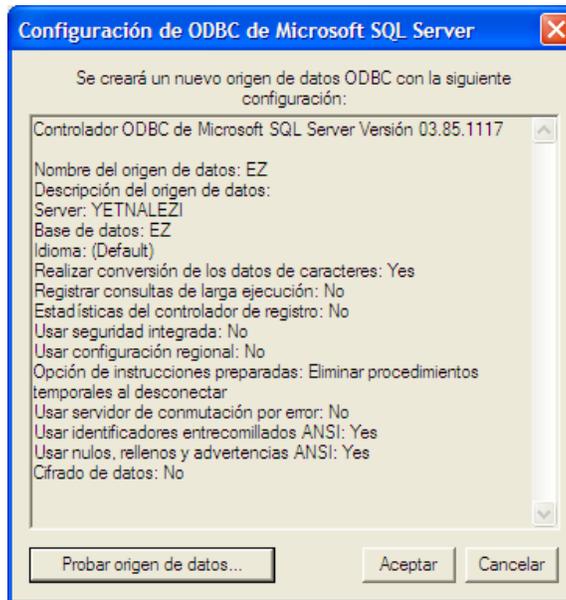


Figura 10. Comprobación exitosa del origen de datos.

Para finalizar, cierre las ventanas. En este momento el sistema EZ-KET está listo para ser utilizado y debe funcionar correctamente. Para mayor información acerca del funcionamiento de EZ-KET revise la documentación del usuario del sistema.